Design and Performance Analysis of a Geographic Routing Protocol for Highly Dynamic MANETs

Kevin Peters

Submitted to the graduate degree program in Electrical Engineering & Computer Science and the Graduate Faculty of the University of Kansas School of Engineering in partial fulfillment of the requirements for the degree of Master of Science.

Thesis Committee:

Dr. James P.G. Sterbenz: Chairperson

Dr. Hossein Saiedian

Dr. Gary Minden

Date Defended

The Thesis Committee for Kevin Peters certifies
that this is the approved version of the following thesis:

Design and Performance Analysis of a Geographic Routing Protocol for Highly Dynamic MANETs

Committee:

Dr. James P.G. Sterbenz: Chairperson

Dr. Hossein Saiedian

Dr. Gary Minden

Date Approved

Abstract

Efficient multi-hop routing has become important for airborne telemetry networks. The highly dynamic nature in these scenarios results in short-lived links. Geographic-based routing has an advantage over topology-based routing to make rapid forwarding decisions based on neighbor and destination position. The AeroRP geographic routing protocol is detailed, which uses a heuristic metric for forwarding decisions that takes transmission range and a neighbor's location and velocity into consideration. The main contributions of this work include detailing and finalizing the routing decision metrics, design, and simulation implementation of AeroRP. The analysis of the simulations shows AeroRP has several advantages over other MANET routing protocols and offers tradeoffs for different performance metrics in the form of different AeroRP modes. Specifically, AeroRP yields higher accuracy than all compared routing protocols and various AeroRP modes can be chosen depending on how packet delivery and delay are prioritized.

Acknowledgments

Thanks to my advisor and chair, Prof. James Sterbenz, for initially sparking my interest in networking and inviting me into the ResiliNets group, which eventually led me to this work. His guidance and support was important in getting me to this point. The PhD students in the ResiliNets group, Justin Rohrer, Abdul Jabbar, and Egemen Çetinkaya were always very helpful and great resources to tap. Specifically, I want to acknowledge Justin's patience in answering the tiniest of questions that I had and his invaluable post processing scripts, and I want to acknowledge Abdul for his handy graphing script and his initial work and continued feedback with developing AeroRP. Hemanth Narra and Yufei Cheng of ResiliNets implemented DSDV in ns-3, which gave me another routing protocol to compare AeroRP against amongst the grand total of two MANET routing protocols that ns-3 had at the time of this writing. Dan Broyles, another graduate student in the ResiliNets group, developed the 3D Gauss-Markov mobility model implementation for ns-3. He was very helpful in responding to my configuration questions and was receptive to making changes to the model such that its parameters could be controlled similar to an aircraft. Finally, many thanks to my family and friends, specifically my wife, Megan, for their patience and understanding as I spent many countless nights and weekends working on this research.

Contents

List of Figures

List of Tables

Chapter 1

Introduction and Motivation

Mobile ad hoc networks (MANETs) are self-configuring wireless networks with no pre-established infrastructure. MANETs have inherent challenges that most common wireless networks do not. These challenges largely stem from the mobility of the MANET nodes, which result in broken links and thus decreasing the connectivity of the network. Routing packets amongst a network in which a specific hop-by-hop path will most likely not persist must be a major consideration by the MANET routing protocol. This is especially true for the highly-dynamic airborne tactical networks that is the focus of this research. These fast-moving nodes create a unique challenge for routing packets when connectivity amongst the nodes is very intermittent and episodic.

Each node in a MANET must act as a router for other nodes' packets. Routing in MANETs can be categorized into topology-based and geographic-based approaches. Topology-based routing protocols have the disadvantage of high overhead due to control messaging amongst the nodes for both proactive and reactive approaches, and routing table maintenance for proactive approaches. Geographic-based routing has the advantage of making forwarding decisions on-the-fly based

on the destination's position and some algorithm to move the packet closer to the destination based on the destination's position and the surrounding nodes. The disadvantage is the extra cost, complexity, and energy use required for nodes to be location aware.

The research that this thesis outlines is to expand and extensively test, tune, and analyze the geographic-based AeroRP routing protocol [1] and compare it with well-known topology-based routing protocols for MANETs. The scenarios explored in this research are bound by the highly dynamic and airborne iNET use cases [7–10] illustrated in Figure 1.1. The main focus of AeroRP is to efficiently route data packets, specifically telemetry data, amongst airborne nodes (ANs) to a ground station (GS). The ANs must use themselves or relay nodes (RNs) as next hops in order for the packets to reach their destination as the AN may not be within transmission range of the GS within a reasonable amount of time.

The different geographic-based routing protocols that are currently available do not seem to take high velocity of the nodes into major consideration. However, when traveling at high speeds (Mach 3.5), the velocity of the node with respect to the destination can be an important consideration for routing. A heuristic metric that takes transmission range and a node's location and velocity into consideration is proposed. This trajectory data is also important for predicting nodes that will not be within transmission range when nodes are moving very quickly in and out of transmission range of one another.

We look at the option of AeroRP storing and carrying packets cite570685 when a better node is not within transmission range, known as the local maximum problem [11]. The reduced delay can make a node a much more effective and desirable ferry when traveling at Mach 3.5 when compared to traditional, slower moving

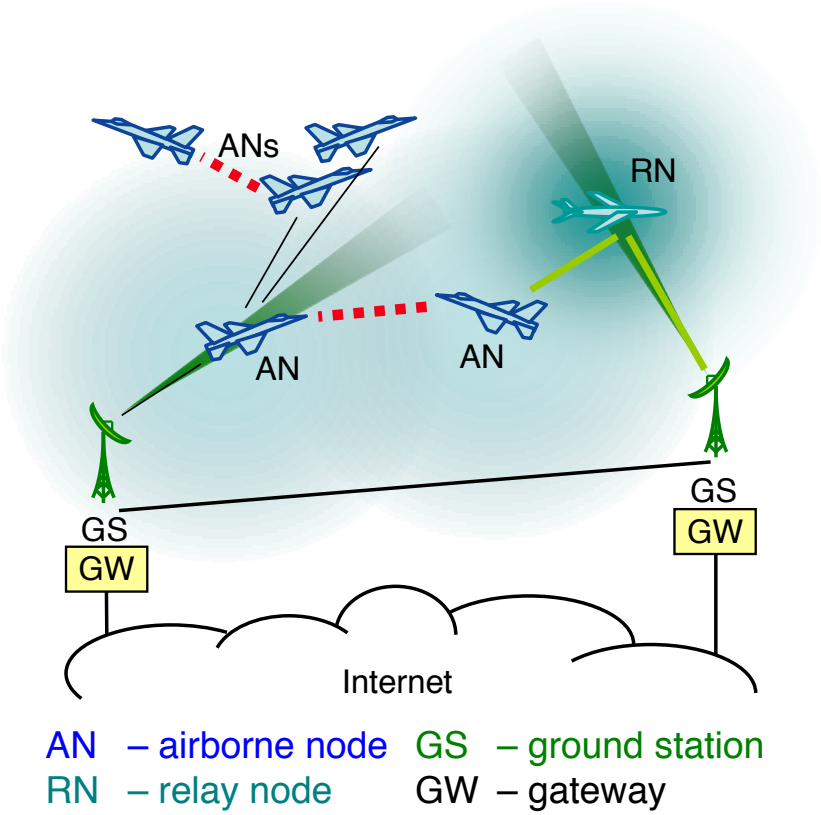

Figure 1.1. Dynamic airborne tactical environment (reproduced from [1])

scenarios. Other geographic based routing protocols have a different secondary backup strategy. One of the major backup strategies implemented by GPSR [4] is face traversal in planar graphs. However, this is not applicable to the 3D environments we are studying and probably not an appropriate approach for such a high speed scenario [12].

1.1 Contributions

The contributions of this thesis are the following:

- Basic validation and performance testing of the new ns-3 network simulator [13]

- Model the AeroRP geographic based routing protocol
  - finalize and confirm the calculations used to make routing decisions
  - detail various modes, including ferrying and beaconless modes
- Implement AeroRP in the ns-3 network simulator
- Analyze the performance of AeroRP against other MANET routing protocols using the ns-3 network simulator

1.2 Thesis Organization

The rest of the thesis is organized as follows. The subject of geograhic-based routing and relevant related research is discussed in Chapter 2. Chapter 3 outlines the simulation model used, validating the model and choosing optimal parameters, how AeroRP makes its routing decisions, how AeroRP is implemented in the simulation model, and the detailed configuration of the simulations. Chapter 4 details the results and analysis of the simulations. Finally, Chapter 5 concludes what has been learned in this research and discussions of areas to focus on for future work.

Chapter 2

Background and Related Work

This chapter presents related research on MANET routing protocols that is the basis for this thesis and is organized as follows: Section 2.1 discusses topology based routing protocols in general as well as the details of the specific topology based routing protocols OLSR, DSDV, AODV, and DSR. Section 2.2 discusses geographic-based routing protocols in general, four popular geographic routing protocols, and the previous work and initial design of AeroRP. Since AeroRP has the option to *ferry* data packets, Section 2.3 details background information on Store and Haul. Finally, Section 2.4 helps to understand other work in which 3D space is considered for routing and Section 2.5 details related work in high-velocity routing since we are working in a highly dynamic aeronautical environment.

2.1 Topology Based Routing Protocols

Topology based routing protocols are typically either link state or distance vector based. One of the main differences between link state and distance vector routing is that distance vector routing only communicates link information with its

direct neighbors whereas link state routing floods the link states to all nodes in the network. Because of this, distance vector routing can be easier to implement and more efficient. However, distance vector routing can have problems like routing loops from which link state approaches do not suffer from. Source routing is another category in which the packet contains the hop-by-hop addresses to the destination in the packet header.

Routing protocols can also be either reactive or proactive. Reactive routing protocols have the advantage of efficiently using network bandwidth only when routes are needed with the disadvantage of the latency to set the route up for the first time. Proactive routing protocols have the advantage of instantaneous route knowledge with the disadvantage of using significant network bandwidth to maintain route updates to all destinations amongst all nodes, which does not scale well for larger node densities.

The following subsections detail the four cannonical topology-based routing protocols; OLSR, DSDV, AODV, and DSR.

2.1.1 OLSR

The Optimized Link State Routing Protocol (OLSR) [14,15] is one of the well-known proactive MANET routing protocols. A simple link-state protocol floods all neighbor links to the entire network. OLSR's main contribution is to take pure link state routing one step further and efficiently distribute its control messages, as opposed to flooding, in order to reduce overhead while still providing optimal routes.

Instead of a node communicating all of its links to the network, it only communicates a subset of these links referred to as multipoint relay (MPR) selectors,

which reduces the size of the control packet. The MPR nodes for a given node are selected by choosing nodes such that all two-hop neighbors will be covered by the one-hop MPR set. It also only floods these control packets to its multipoint relays, which reduces the traffic when compared to flooding. The non-MPR nodes for a node can overhear the control messages but do not retransmit.

`HELLO` beacons are used for nodes to determine their one-hop neighbors as well as two-hop neighbors in order to construct their MPR selector set. Since the `HELLO` beacon contains the list of neighbors that a node can hear, this allows nodes within transmission range of the `HELLO` beacon to learn about two-hop neighbors. The `HELLO` beacons also indicate the status of the links with its neighbors, including whether or not a given neighbor is an MPR for that node.

Each node broadcasts Topology Control (TC) messages via its MPRs. The MPR selector set is contained in these TC messages and allows the nodes in the network to build its topology table. This TC information coupled with the neighbor table collected from the `HELLO` beacons is enough for a given node to calculate the next hop for a given destination and can thus build its routing table.

2.1.2 DSDV

Distance-Sequenced Distance-Vector (DSDV) [16] is also a proactive routing protocol; but using the distance vector algorithm rather than link state. In DSDV, each node periodically transmits updates to its direct neighbors. Depending on the situation and in order to efficiently use the network bandwidth, the updates can be in the form of full dumps or incremental dumps. A full dump is the entire routing table and an incremental dump is just the routing data that has changed since the last update. Each node's routing table is a list of all known nodes, the

number of hops (distance) to each node, and the next-hop node address, to take for a given destination node.

One of the main contributions of DSDV compared to its wired distance vector predecessors is a sequence number that is associated with all routing entries and thus all updates that are made. A sequence number allows a node to distinguish between older and newer routing data. Specifically, a node can determine if a link is still broken based on the sequence number indicating the link was broken and the sequence number of a routing update indicating the link is not broken.

DSDV also keeps track of the difference in time between routing table updates and sends a given update based on how frequent or infrequent the routing data is changing. For instance, if the routing information being sent to a node for a given destination is rapidly changing, that node will not send updates as frequently in order to give the route sufficient time to settle. This is referred to as the settling time so that network bandwidth is not wasted on a rapidly changing route. Note that the discovery of broken links are always immediately advertised.

2.1.3 AODV

Like DSDV, Ad-hoc On-Demand Distance Vector (AODV) [17] is also a distance vector based routing protocol that uses sequence numbers to distinguish fresh routing data from stale routing data. However, a key difference is that AODV is a reactive protocol whereas DSDV is a proactive protocol. In AODV, a node does not proactively seek routes to all destinations and only tries to discover a route to a destination when the two nodes need to communicate.

A path to a destination is discovered by the source node broadcasting a route request (`RREQ`) packet that contains the source and destination address along with

sequence numbers and other bookkeeping details. If the neighbor who receives the broadcast RREQ does not have a route for the destination, it increases the hop count and rebroadcasts the RREQ. Duplicate RREQs are dropped.

As the RREQ packet makes it way from the source to the destination, the intermediate nodes store information in order to get packets back to the source, known as the reverse path setup. When a route is found, the node uses this reverse path to send a route reply packet (RREP) back to the source. Each node that gets this RREP sets up a forward pointer to the node that the RREP came from in order to establish the route.

Various timers are used to ensure that the stale data is purged from route entries. Similarly to DSDV, old route entries can be replaced by newer route entires with lower hop counts, indicated by sequence numbers. Broken links are advertised as infinite hop counts in the RREP amongst the nodes. A node with a broken link in its route would need to start the route discovery process over again with a new RREQ packet.

2.1.4 DSR

Dynamic Source Routing (DSR) [18, 19] is neither distance vector nor link state based. However, it is a reactive protocol like AODV in which a route is only sought out when needed thus reducing control overhead. In DSR, the data packets contain the hop-by-hop instructions to the destination in the packet header. Each node that receives the packet knows the next hop by examining the *source route* in the packet header.

A node that does not have a path in its routing table broadcasts a RREQ packet. The RREQ is propagated through the network, using a sequence number to prevent

duplicates and updating the route the packet has traveled until the destination is reached. When the destination is reached, a `RREP` is sent back to the source with the source to destination hops. DSR relies on nodes that detect link errors to send a `RERR` to the originator so that its routing table can be updated appropriately.

The nodes can also update their routing tables based on data traffic or `RREP` packets that they are forwarding or overhearing from their neighbors. The packets contain the end-to-end path for a specific source to destination, which neighbors can store for future use. Other optimizations exist like piggybacking data on `RREQ`, `RREP`, and `RERR` packets.

2.2 Geographic Based Routing Protocols

The various geographic routing survey papers [2, 20, 21] break down different geographic forwarding decisions into MFR (most forward with radius r), NFP (nearest with forward progress), and compass. MFR is the most intuitive and forwards the packet to the node, which makes the most forward progress between the source and destination. NFP forwards the packet that is closest to the current node and is closer to the destination. NFP is meant to reduce packet collision when compared with MFR by making shorter hop routing decisions. Compass forwarding chooses a node that is closest to an imaginary line drawn between itself and the destination and thus taking the trajectory of the nodes into more consideration. These different approaches are illustrated in Figure 2.1.

Note that the concept of location services is not applicable to this research as it is assumed that all nodes know their position with the aid of some positioning system like the Global Positioning System (GPS).

There are many popular geographic routing protocols, including DREAM,

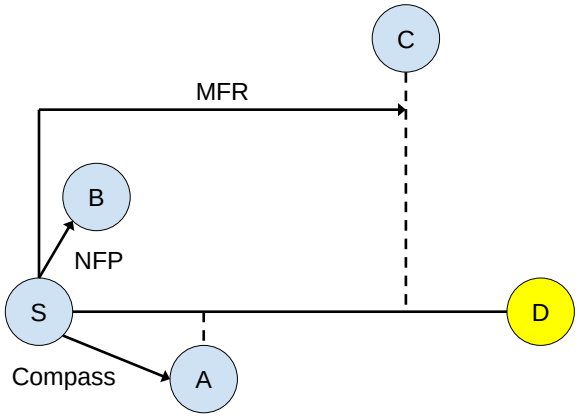

Figure 2.1. Geographic routing approaches (adapted from [2])

LAR, GPSR, and SiFT, each of which are detailed in the following subsections. The last subsection describes the original work on the AeroRP routing protocol.

2.2.1 DREAM

The Distance Routing Effect Algorithm for Mobility (DREAM) [3] was one of the first popular geographic routing protocols adopted. In DREAM, the frequency at which location information is shared amongst the nodes is based on how far apart the nodes are and how fast the nodes are moving. The control message contains the node identifier and coordinates (no trajectory information is shared). The further apart a given node is from another node, the less frequent location information needs to be shared. A node must share its location more often the faster it is moving. DREAM optimizes the frequencies of its control messages based on these premises.

Based on the location information that the nodes collect based on the control messages, DREAM moves the data packets with no pre-established route to nodes that it knows are towards the direction of the destination. A packet is sent amongst a node's one-hop neighbors by sending to all of the neighbors that lie

within a wedge that originates from the sender and opens up to the possible distance that the receiver travels in a given unit of time. The maximum velocity is known and used to calculate the possible distance that the destination could move. This wedge is illustrated as θ in Figure 2.2 where S is the source and D is the destination. Due to the nature of the wireless medium, all one hop neighbors will hear the packet. However, the sender and receiver know if a node is in the wedge based on previously exchanged location information. This process is repeated at each hop with an undefined recovery mechanism if there are no one-hop neighbors within the wedge.

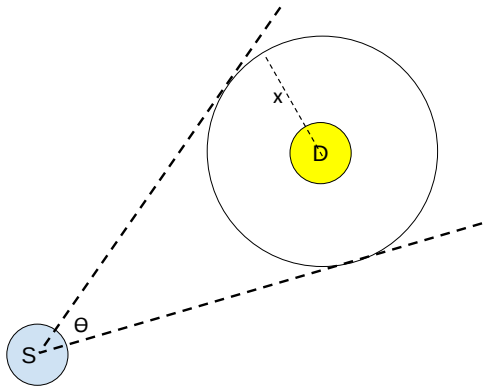

Figure 2.2. x is the distance that D can travel (adapted from [3])

2.2.2 LAR

Like DREAM, Location-Aided Routing (LAR) [22] was amongst some of the first adopted routing protocols to take location information into consideration when routing in MANETs. LAR uses the same concept from DREAM of the wedge as illustrated in Figure 2.2 and refers to it as the *request zone*. However, unlike DREAM, it uses this request zone to send route requests as opposed to data packets. Packets are flooded to those nodes that are in the request zone.

LAR degrades to flooding if forwarding packets to nodes within the request zone do not reach the destination.

In order for LAR to work, nodes must know if they are in the request zone so they can either drop or continue to flood the packet. The LAR authors propose two different schemes for a node to determine if it is in the request zone. The first scheme consists of the sender sending a route request that contains the coordinates of a rectangle that contains the request zone. A node that receives this route request will discard it if it is not within the rectangle and forward it on if it is. Once the route request reaches the destination, it replies with the route reply message. The second schema does not explicitly define the request zone when sending the route request but instead forwards the packet based on the distance the sending node is from the destination, which is included in the route request.

2.2.3 GPSR

Greedy Perimeter Stateless Routing (GPSR) [4] forwards data packets based on a greedy heuristic and then uses perimeter routing if the greedy approach is not possible. A beaconing mechanism is used to share location information with one-hop neighbors as well as piggybacking location information on actual data packets. The greedy approach is straightforward and illustrated in the choice of node C as the next hop in Figure 2.1. However, packets cannot always be forwarded with this approach even when other paths exist as illustrated in Figure 2.3.

So, GPSR takes another approach, perimeter routing, when greedy forwarding does not work. Well known techniques for traversing the perimeter of a planar graph are used to forward the packet, as illustrated in Figure 2.4, until greedy forwarding can resume.

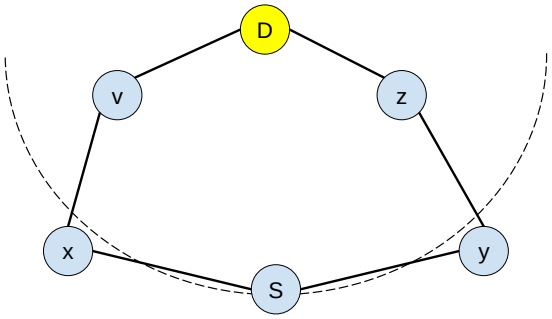

Figure 2.3. S cannot choose x or y because S is closer to D (adapted from [4])

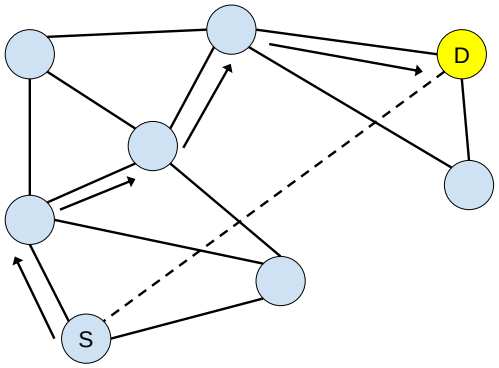

Figure 2.4. Perimeter mode (adapted from [4])

2.2.4 SiFT

Simple Forwarding over Trajectory (SiFT) [23] is different than the other geographic routing protocols discussed as it not only shares position information but trajectory information as well in the data packets. The sender broadcasts data packets and leaves the routing decision up to the receiver based on its position and trajectory, thus eliminating the need for control packets. When a node receives a packet, it starts a timer, which is a function of the distance from the last sender and the distance from the trajectory such that the node farthest from the

last node and closest to the destination will have the shortest timer. The packet is discarded if it receives another copy of the packet before the timer expires. Otherwise, the packet is broadcasted again when the timer expires.

SiFT has other unique characteristics such as multicast via *trajectory trees* as well as the ability to optimize the timers based on interacting with the MAC layer.

2.2.5 AeroRP

The original AeroRP routing protocol was first introduced in [1] and described in [24]. The protocol is meant to handle situations in which mobile nodes are moving at speeds of up to Mach 3.5, which results in a highly dynamic topology and very intermittent connectivity amongst the nodes. This is achieved by making per-hop routing decisions to move the data packet closer to the destination without knowledge of a full end-to-end path.

The first phase of AeroRP is *neighbor discovery* in which the nodes determine their neighbors using various mechanisms. One of those mechanisms is *active snooping* in which a node that is not transmitting overhears other nodes' transmissions and uses the data from the overhead transmissions to store trajectory information regarding an overheard node. `Hello` beacons are another mechanism in which nodes explicitly advertise their presence at some regular interval. The final mechanism consists of the nodes receiving updates from a ground station that contains trajectory information predicted by the mission plan.

The second phase of AeroRP is *data forwarding* in which a node must decide the next hop for a data packet. The location of the destination is known by all nodes ahead of time, and a neighbor table is maintained by each node that is updated based on the mechanism used in the first phase. The node uses this

information to choose the next hop for each data packet. This is done by choosing the neighbor that has the smallest *time to intercept* (TTI), which is a metric that indicates the time it will take for a node to be within transmission range of the destination if it continues on its current trajectory. Assume that node n_0 wants to send a data packet to the ground station D. Assume that the transmission range of all nodes is R. The TTI for each node is calculated as:

$$\text{TTI} = \frac{\Delta d - R}{s_{\text{d}}} \tag{2.1}$$

in which Δd gives the euclidean distance between the current location of a potential node and the destination node n_{d}, and s_{d} is the component of the actual speed of the potential node in the direction of the destination. The actual calculation for s_{d} is not given in [24].

Preliminary testing with the ns-2 simulator [25] compared AeroRP with AODV and DSDV in high speed scenarios. AeroRP significantly outperformed both AODV and DSDV in both packets received and protocol overhead.

2.3 Store and Haul

The AeroRP routing protocol defined in this thesis has the option to ferry packets, otherwise known as Store and Haul (S&H). S&H uses node mobility to physically carry data to regions of MANETs where communication is not possible due to connectivity limitations, interference, or eavesdropping [26]. S&H is almost universally classified as a Delay Tolerant Networking (DTN) strategy [27] or opportunistic networking [28].

Since the proposed concept of S&H, there have been several variations pro-

posed. It has been referred to as Message Ferrying (MF) [29,30], Store and Carry Forward (SCF) [31], and Store and Forward [32,33]. In these variations, different kinds of strategies were proposed that involved random ferrying, dedicated ferries, differentiated services across ferries, as well as different kinds of ferries. For the purposes of this research and the AeroRP routing protocol, any node can be a ferry if certain conditions are met, which are discussed in detail in Section 3.2.4. The ferry and S&H terminologies are treated as equivalent in this research.

Geographic based routing in conjunction with S&H is discussed in [11] and [34]. However, none of the surveyed research touches on the fact and implications that reduced ferry delay due to Mach 3.5 speeds can make a node a much more effective and desirable ferry. Although Mach 3.5 reaches nowhere near the speed of light that data can be transferred to another node, ferrying the data in such a high speed manner may be desirable in certain scenarios whereas traditional slower speed ferrying would have been unacceptable.

2.4 Routing in 3D Space

Most geographic routing protocols assume that the nodes are deployed in 2D space. Since the focus of this research is on airborne nodes, the interest is in geographic routing in 3D space of which there is not a definitive approach that guarantees packet delivery [12]. The Gauss-Markov mobility model that was implemented in ns-3 for this research and detailed in Section 3.4.2 allows for travel in 3D space [35]. Although the speed component s_{d} of the heuristic used to make routing decisions in AeroRP only considers velocity in 2D space (Section 3.2.1), the distance used for the TTI calculation (Section 3.2.2) and expiring neighbors based on where they will be in the future (Section 3.2.3) both consider 3D space.

It's important to note that several recovery modes are well known for when the local maximum problem is reached in 2D space such as face traversal in planar graphs discussed in Section 2.2.3. However, there is not a well known or practiced solution for guaranteed delivery for 3D space since the concept of planar graphs is not applicable for 3D space. A randomized walk approach for recovering from the local maximum problem in 3D space has been discussed and proven [12]. A spherical coordinate tree to recover when a greedy path is not available is another approach [36].

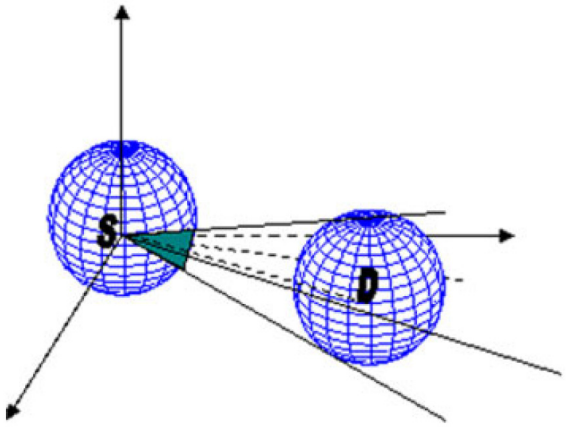

Figure 2.5. Geocasting region is the cone (reproduced from [5])

The unique approach of geocasting in 3D in the 3D geographic routing (3DGR) [5] algorithm helps illustrate the subtle differences in 2D versus 3D routing considerations. The idea is to send a control message for a destination path with some initial geocasting angle. If a path is not found, the geocasting angle is increased and the search for a path is repeated. The geocasting region is illustrated in Figure 2.5. A specific example comparing GPSR, discussed in Section 2.2.3, and 3DGR is helpful to understand how 3DGR not only overcomes the local maximum problem by using the geocasting approach, but how a better path than GPSR would calculate is chosen. Imagine that node S needs to send a packet to node D in Figure 2.6.

There is no greedy path so both GPSR and 3DGR would need to degrade to their recovery mode. GPSR will use its perimeter mode, which will result in a path A-B-C-E-F-G-H-I-D. However, with 3DGR, it will increase its geocasting angle towards the destination until it hears about a path from both A and F. A is the more expensive path so F will be chosen resulting in the S-F-G-H-I-D optimal path.

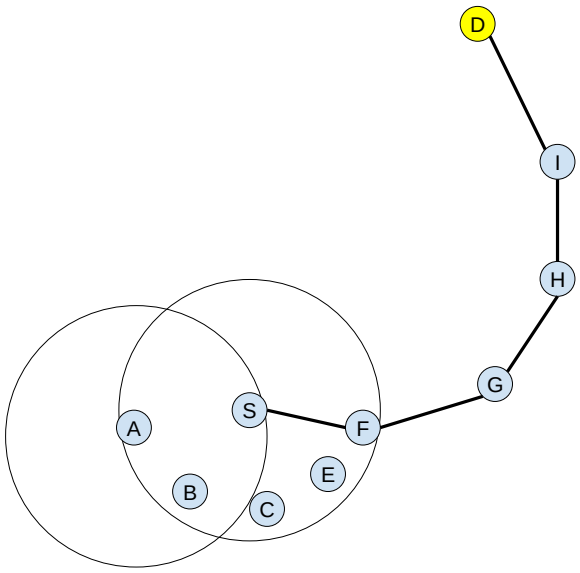

Figure 2.6. Routing example (adapted from [5])

2.5 Routing Amongst High Speed Nodes

Unmanned aerial vehicles and geographic routing has been simulated at 25 m/s [37]. However, the aerial vehicles being simulated for AeroRP are traveling much faster at 1200 m/s. A top speed of around 20-50 m/s seems to be typical amongst most of the surveyed papers that indicate the speeds used for testing various geographical routing protocols [11, 34, 38–42].

Although most MANET routing protocols are focused on lower speed environ-

ments than the focus of this, there are a few routing protocols in the literature with aeronautical environments in mind. ARPAM [43] is a hybrid AODV [17] protocol for commercial aviation networks that utilizes the geographic locations to discover the shortest but complete end-to-end path between source and destination. Multipath Doplar Routing (MUDOR) [44] takes relative velocity into consideration as well as the Doppler shift to measure the quality of a link. Anticipatory Routing [45] tracks highly mobile endpoints that reach the reactive limit in which the speed of the nodes is comparable to the time it takes for the location tracking to converge upon the position of the node. Spray Routing [46] involves unicasting a packet a specific *depth* away from the destination in which the packet is then *sprayed* or multicasted to a controlled *width* or number of levels of neighbors. Spray Routing specifically has highly mobile endpoints in mind. The authors even test with high speeds in which the throughput approaches 0 around 250 m/s. However, none of these approaches mention such speeds as high as Mach 3.5 in which rapidly varying connectivity is a major consideration.

Chapter 3

Simulations

This chapter explores some basic ns-3 performance testing and validation that was required to obtain acceptable results from the simulator. The specific details of AeroRP, including how it makes its routing decisions, the flow of the routing protocol, and how it is actually implemented in ns-3, are discussed. The simulation configurations are also detailed, including how the ns-3 simulator is setup, how the routing protocols are configured, and the various mobility models that are used for the simulations.

The following sections detail ns-3 validation and benchmarking in Section 3.1, the current AeroRP implementation in Section 3.2 and Section 3.3, and specific configuration of the simulations in Section 3.4.

3.1 Validating and Benchmarking ns-3

Initial testing with the ns-3 network simulator showed that the throughput of the simulations were very sensitive to various configuration changes. One has to find a good balance between having a high enough transmission power such that a

node can communicate with its neighbors that are within some reasonable distance for a given scenario and a low enough transmission power such that the nodes are not interfering with each other at a high rate. We want to avoid the *parking lot problem* [47], which is the breakdown of communication due to self jamming and signal overload in the case of very dense network nodes. Simple wireless channel models are not provided in ns-3 in which it is easy to set transmission range. Other testing, consideration, and configuration is required for longer transmission ranges, frame preamble size, sharing the channel, and discovering other optimal values like transmission rate. The following sections detail various performance testing and tunings made to fit the aeronautical use cases being researched in this thesis.

3.1.1 Short Transmission Range

In order to get a basic understanding of performance in ns-3, we first start with a simple scenario of one node sending to another node that is 500 m away. This is a relatively short transmission range compared to the transmission ranges of 27800 m for the aeronautical use cases. Experiments were conducted with various transmission powers. The details of the experiment are defined in Table 3.1. OLSR was chosen as the routing protocol for these performance tests as it is the most mature MANET routing protocol in the ns-3 simulation suite. The receiving rate for each transmission power and sending rate experiment was averaged over the simulation run to give Figure 3.1 in which each different line is a different transmission power. One can see that just going from 15.1 dBm to 15.2 dBm results in more than 1 Mb/s improved throughput for the higher sending rates. Note that this specific ns-3 test yields a throughput of 4.5 Mb/s at 6 Mb/s sending

rate for 802.11b in 11 Mb/s mode.

Table 3.1. ns-3 benchmarking simulation parameters

Parameter	Value
Routing	OLSR
Warmup time	30s
Simulation time	100s
Link layer	wifib-11mbs
Packet size	1000 bytes
Sending rate	6000 kb/s (6 Mb/s)
RTS/CTS?	No
Packet fragmentation?	No
Mobility model	Static
Propagation loss model	Friis
Transport protocol	UDP

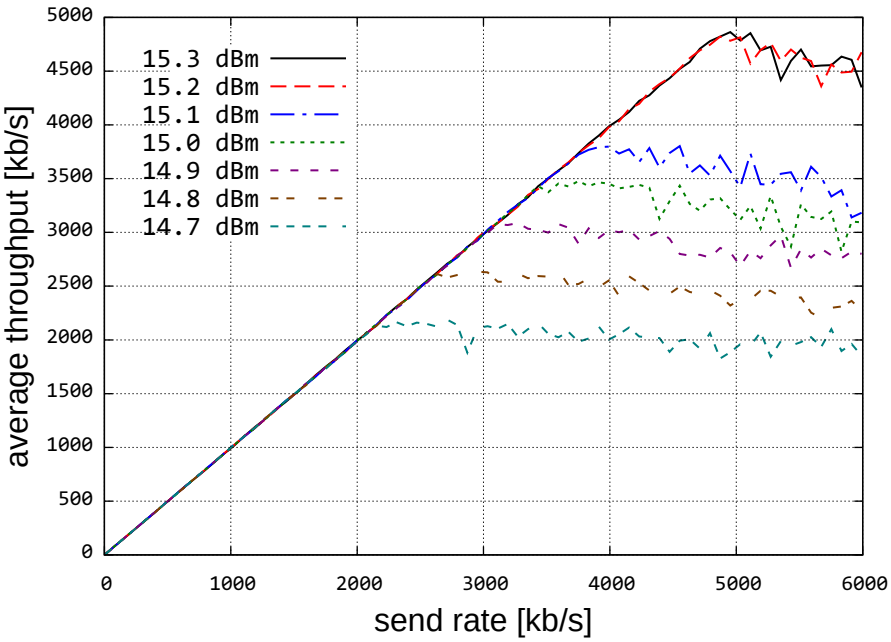

Figure 3.1. One non-mobile node sending to another node

The optimal transmission power of 15.2 dBm derived from testing allows us to better understand how ns-3 is simulating at the lower physical and MAC layers, including the propagation loss model being used. In our case, we are using the

Friis propagation loss model [48], which is shown in the following equation:

$$\frac{P_r}{P_t} = G_t G_r \left(\frac{\lambda}{4\pi R}\right)^2 \tag{3.1}$$

In which P_r is receive power, P_t is transmit power, G_t is transmit antenna gain, G_r is receive antenna gain, λ is the wavelength, and R is the range. Transmit and receive antenna gain are ignored in the simulation. From testing of the simple 500 m scenario, we know all of the values except for the receive power P_r sensitivity required to get this performance. The following are the known values for the 500 m scenario:

$$\begin{aligned} P_t &= 15.2 \text{ dBm} = 10^{\frac{15.2 \text{ dBm} - 30}{10}} = 0.033113 \text{ W} \\ \lambda &= 2.4 \text{ GHz} = 0.12491 \text{ m} \\ R &= 500 \text{ m} \end{aligned}$$

From Equation 3.1, we can calculate the receive power P_r as follows:

$$\begin{aligned} P_r &= P_t \left(\frac{\lambda}{4\pi R}\right)^2 \\ &= 0.033113 \text{ W} \left(\frac{0.12491 \text{ m}}{4\pi 500 \text{ m}}\right)^2 \\ &= 1.308679 \times 10^{-11} \text{ W} \\ &= 10 \log_{10} \left(1000 \times 1.308679 \times 10^{-11} \text{ W}\right) \\ &= -78.832 \text{ dBm} \end{aligned}$$

Knowing the receive power P_r sensitivity in ns-3 will help in establishing ballpark transmission power P_t calculations for different transmission ranges.

The simple scenario from Figure 3.1 was further expanded such that two nodes were moved out of transmission range of one another at 1000 m. However, a new node was introduced in the middle to serve as a next hop. Again, the parameters in Table 3.1 were used for the experiment. The receiving rate for each transmission power and sending rate experiment was averaged to give Figure 3.2 in which each different line is a different transmission power. One can see that just going from 15.1 dBm to 15.2 dBm results in more than 0.5 Mb/s improved throughput for the higher sending rates. As expected, the throughput is roughly half that of the first experiment (Figure 3.1) because of the hop in the middle that is receiving and sending data in between the two nodes.

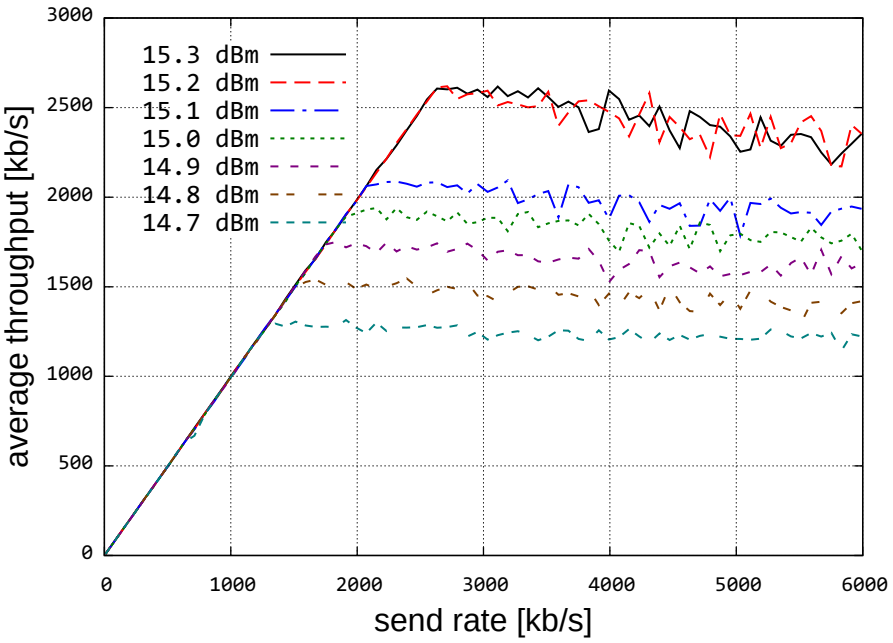

Figure 3.2. One non-mobile node sending to out of range node

The next basic performance test was to observe the effect of the *parking lot problem* [47] by increasing the number of nodes that are sending. The node arrangements are shown in Figure 3.3. The number of the nodes in the figure

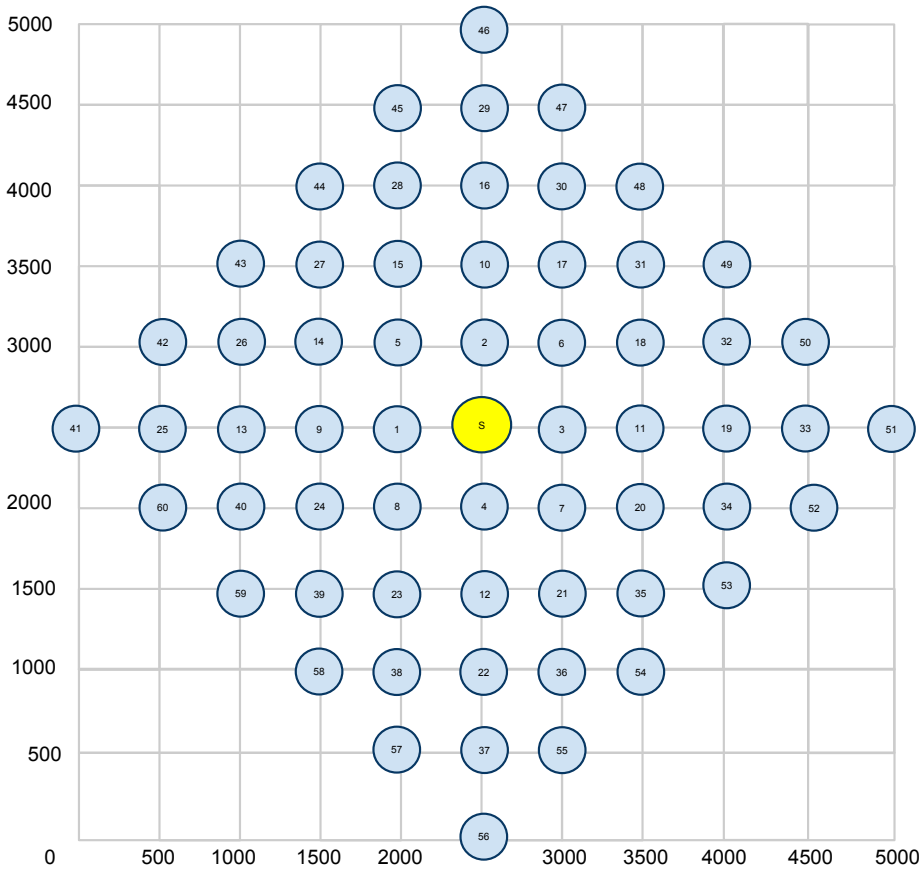

Figure 3.3. Star pattern of nodes

indicates the order they were brought online for each subsequent simulation. All nodes are sending to the same S node in the center. The same parameters from Table 3.1 were used with a couple of exceptions. From the previous two and three node experiments, we know that 15.2 dBm seems to be the optimal transmission power for a distance of 500 m. So, this test was run only with a transmission power of 15.2 dBm and a constant sending rate for each node of 6 Mb/s. The results of the test are shown in Figure 3.4 in which the x-axis indicates the node density. The figure shows that the expected throughput of more than 4.5 Mb/s is seen for smaller node densities but the throughput degrades, eventually to around

2.5 Mb/s, as the node density increases to 61 causing more interference and less throughput.

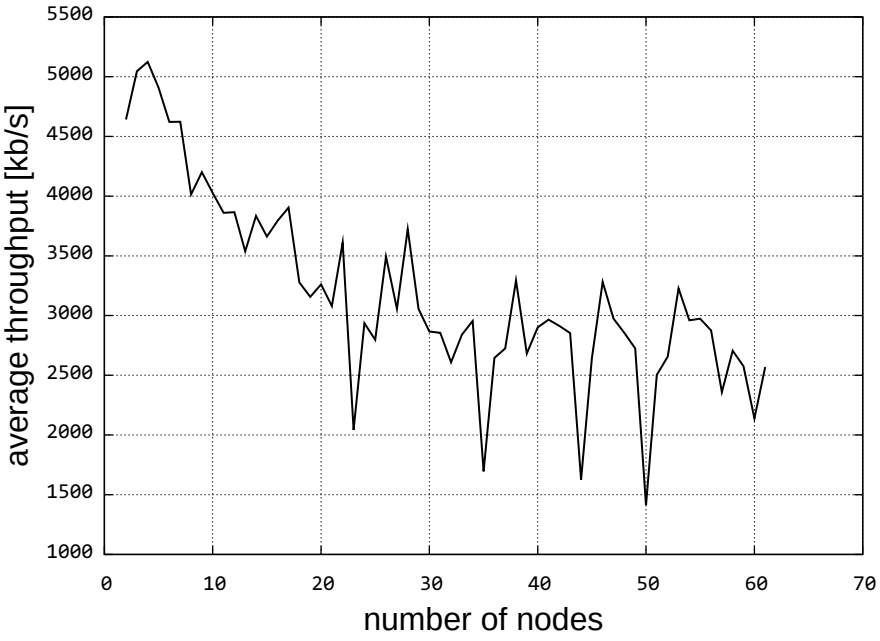

Figure 3.4. Throughput for non-mobile 500 m star pattern

3.1.2 Long Transmission Range

The next logical step is to take these tests and extend them from 500 m transmission ranges to the 27800 m transmission ranges that are required for the aeronautical use cases. However, with the simple scenario of one node sending to another node at 6 Mb/s that is 27800 m away, it was quickly seen that the throughput was drastically lower at 200 kb/s when compared to the 500 m scenario, which is 4.5 Mb/s. This occurred even for very high transmission powers P_t such as 350 dBm. Further testing, illustrated in Figure 3.5, indicated that the throughput was drastically dropping off between 3800 m and 3900 m and consistently staying at around 200 kb/s all the way up to 27800 m.

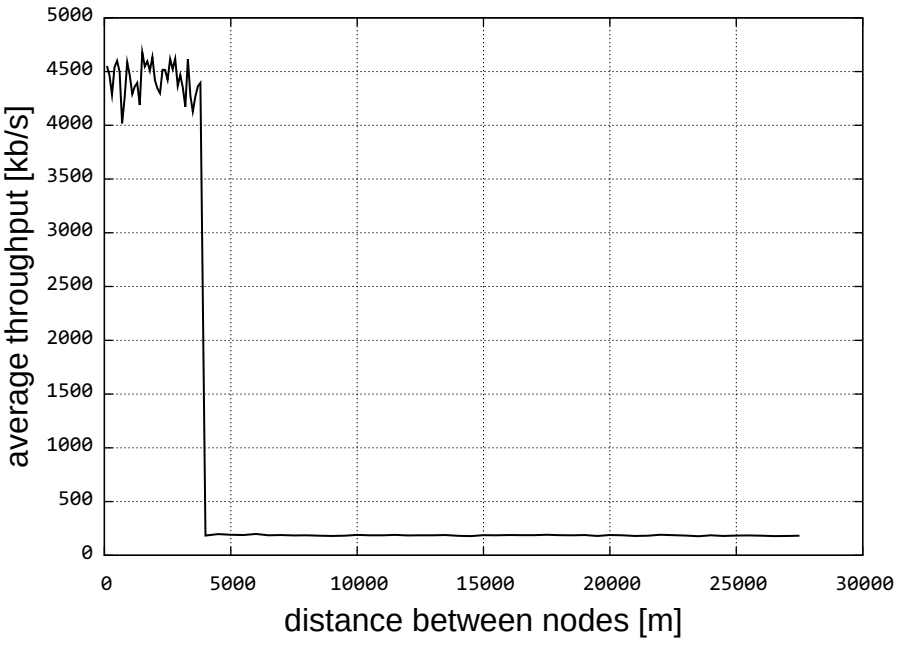

Figure 3.5. Throughput for default ns-3 as distance increases

Based on the 802.11 specification [49], the clear to send (CTS) and ACK timeout values are calculated by adding the short interframe space (SIFS) time, slot time, receive delay time, and the round trip propagation delay between the sender and receiver. For 802.11b, the SIFS time is 10 microseconds, the slot time is 20 microseconds, and the receive delay time is 304 microseconds. The default ns-3 calculations for the CTS and ACK timeout values reflect this except that the round trip propagation delay is calculated by dividing 1000 m by the speed of light. The solution is to set the round trip propagation delay based on the maximum distance between two nodes, which is 27800 m for the aeronautical use cases. Fixing these calculations resulted in drastically better throughput. With this enhancement, Figure 3.6 shows that 4.0 Mb/s throughput can be achieved at distances of 27800 m with a transmission power of 51 dBm. Based on Equation 3.1, we get the following transmission power for 27800 m range, which is in line with

what the actual simulation yielded:

$$
\begin{aligned}
P_t &= \frac{P_r}{\left(\frac{\lambda}{4\pi R}\right)^2} \\
&= \frac{1.308679 \times 10^{-11} \text{ W}}{\left(\frac{0.12491 \text{ m}}{4\pi 27800 \text{ m}}\right)^2} \\
&= 103.26 \text{ W} \\
&= 10 \log_{10} (1000 \times 103.26 \text{ W}) \\
&= 50.139 \text{ dBm}
\end{aligned}
$$

The next test is the star pattern as illustrated in Figure 3.3 except that the nodes

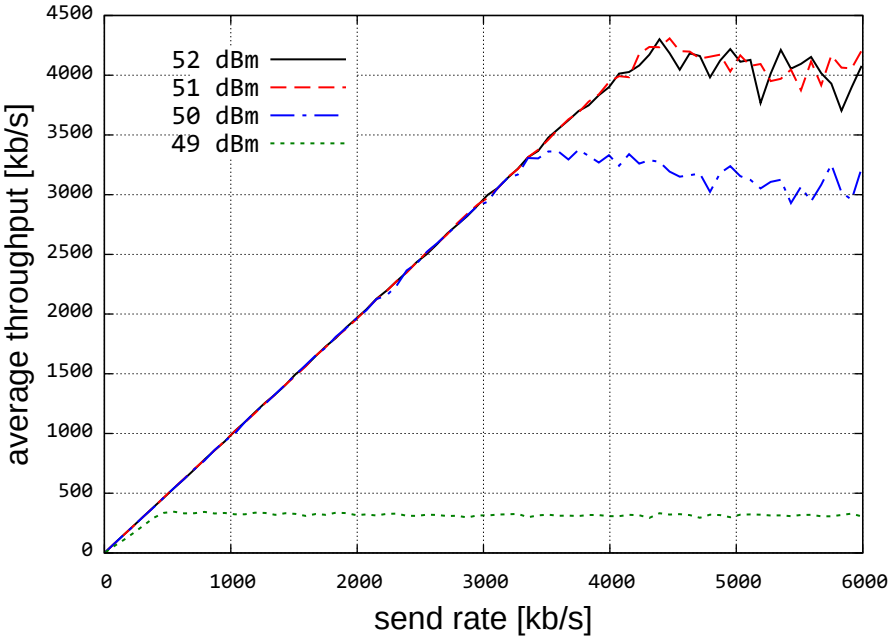

Figure 3.6. Throughput for 27800 m with ACK and CTS enhancement

are separated by 27800 m instead of 500 m. With the ACK and CTS enhancement, almost 2.0 Mb/s throughput was achieved as the simulation approached 61 nodes sending to the sink in the middle as illustrated in Figure 3.7.

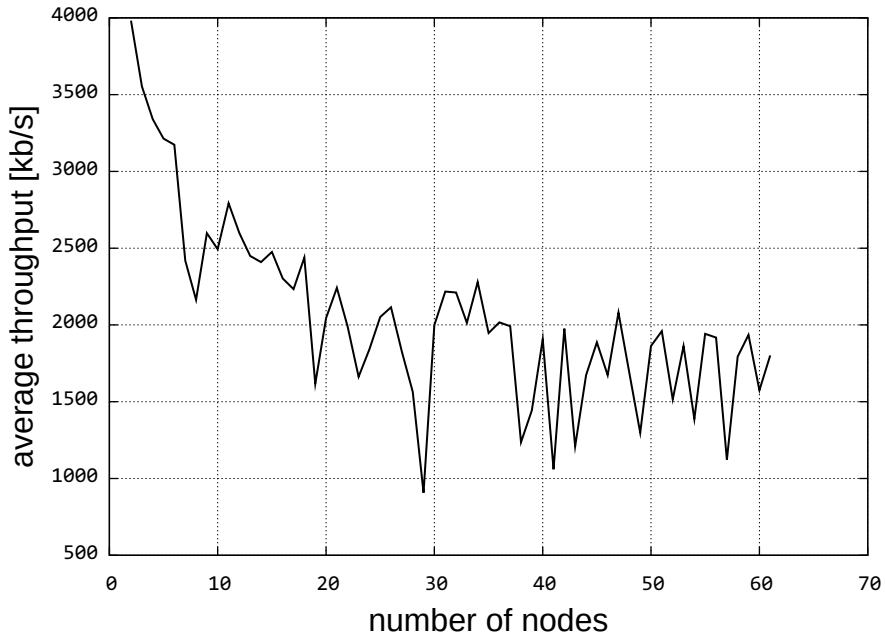

Figure 3.7. Throughput for non-mobile 27800 m star pattern

3.1.3 Long vs. Short PLCP Preamble

Currently, ns-3 uses the long physical layer convergence procedure (PLCP) preamble. This preamble is required for every sent frame. The preamble size effects how long it takes to transmit a packet, including data, ACK, and CTS packets. Section 19.3.3.1 of [49] details 802.11b's support for both a long and a short PLCP preamble. Not only is the short preamble smaller in size and thus does not take us much time to transmit, the short preamble header is submitted at a faster rate of 2 Mb/s whereas the long preamble header is sent at 1 Mb/s.

Since the short preamble mode can theoretically increase throughput at higher data rates, the MAC layer of ns-3 was enhanced to toggle between both the short and long PLCP preamble in order to test possible performance improvements. Testing was done with one sender and receiver separated by 27800 m with the same configuration as described in Section 3.1.2 with a transmission power of

51 dBm. Figure 3.8 shows that using the short preamble can increase throughput more than 0.5 Mb/s when the sending rate reaches 4 Mb/s and higher. The same star pattern test from Section 3.1.2 was also run with both the long and short preamble. Figure 3.9 shows that the short preamble outperforms the long preamble for most node densities. However, there is not a significant difference observed with this specific test.

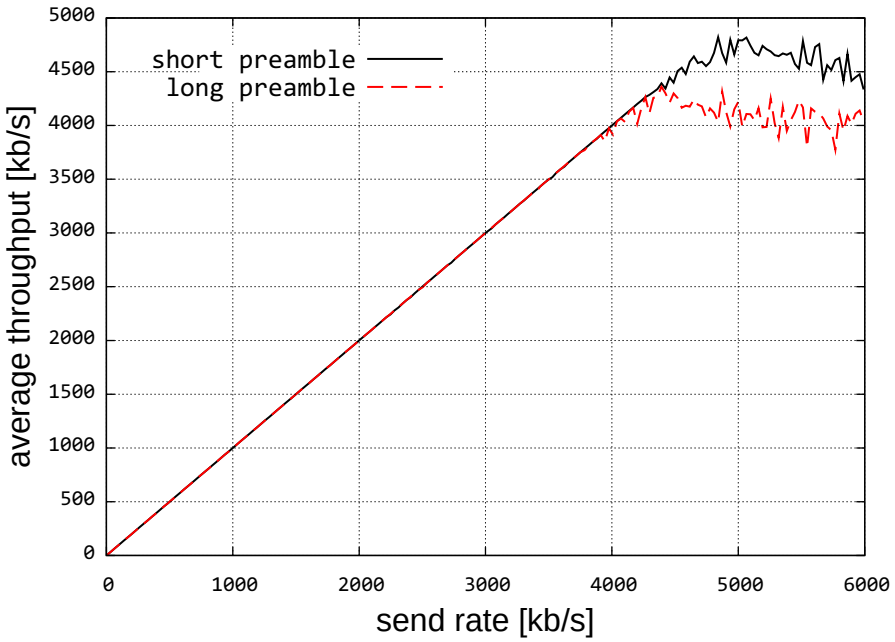

Figure 3.8. Short and long PLCP preamble for 1 sender and receiver

3.1.4 Sharing the Channel

Another consideration when simulating nodes at longer transmission ranges is to fairly share the channel when two or more nodes are overlapping in the channel. With longer transmission ranges, it will be common for nodes to be of widely varying distances from the next hop or final destination.

To investigate the channel and thus throughput shared between sending nodes,

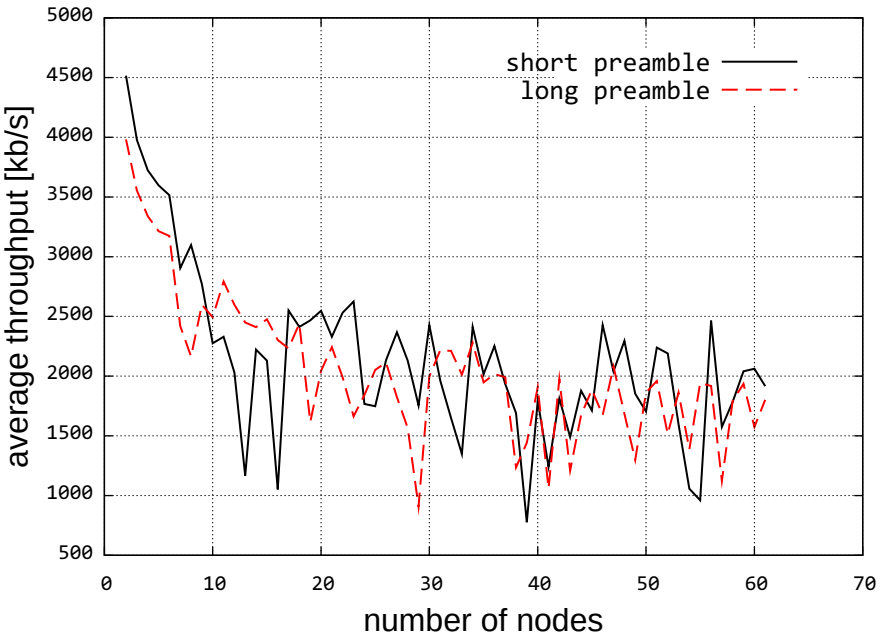

Figure 3.9. Short and long PLCP preamble for star pattern

we test with three nodes in which there are two sending nodes, one on each side of a single receiving node. One of the sending nodes is fixed at 100 m from the destination and the other sending node is moved farther and farther away from the destination. The transmission power is set very high at 350 dB for this experiment so that transmission range is not a concern. Testing with the default ns-3 simulator with the ACK and CTS timeout enhancements discussed in Section 3.1.2 and the long PLCP preamble discussed in Section 3.1.3 showed that the node closest to the source monopolizes the channel as the other node moves away as illustrated in Figure 3.10.

This phenomenon occurs due to the carrier sense multiple access with collision avoidance (CSMA/CA) DCF (distributed coordination function), which 802.11 uses [49]. Nodes will wait for the channel to be idle for a certain duration, DCF interframe space (DIFS), before transmitting. Nodes will not transmit if the

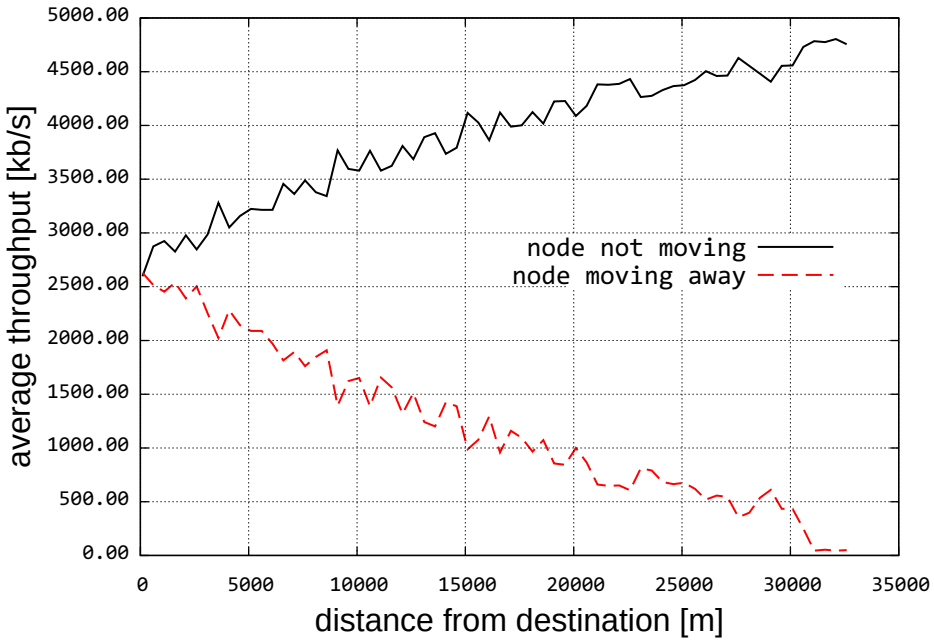

Figure 3.10. Nodes sharing channel in default ns-3

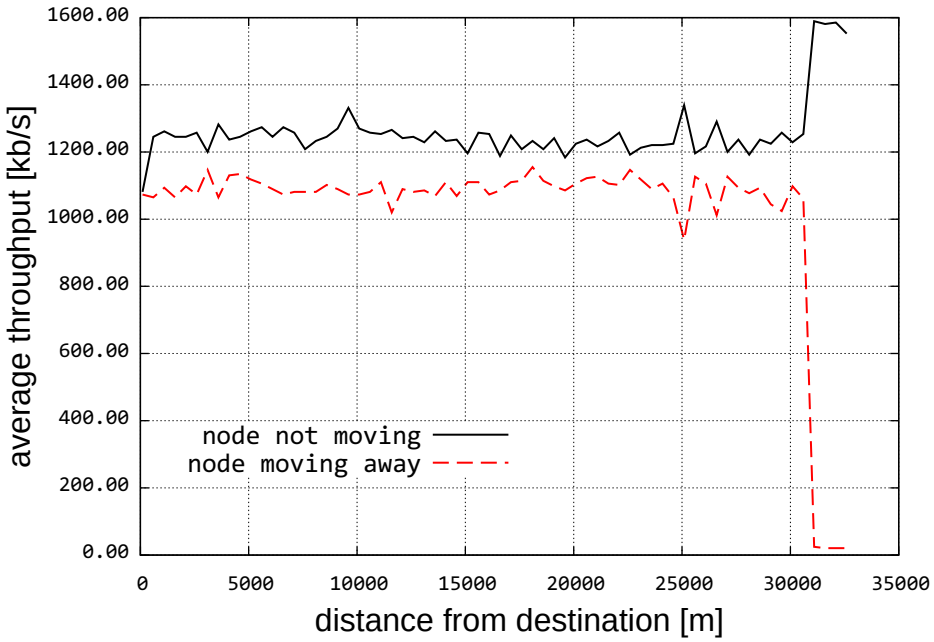

Figure 3.11. Nodes sharing channel when increasing DIFS slot size

channel is busy during the DIFS interval. Increasing the slot size associated with the DIFS interval to account for the longer propagation delay in a scenario in which the transmission ranges are 27800 m allows nodes further from the destination to be able to share the channel with nodes that are closer. This gives the node that is further away more of a chance to use the channel instead of the closer node monopolizing the channel because it always detects the channel is idle quicker than the node further away. The slot size was increased from the default 802.11b of 20 microseconds to 20 microseconds plus the round trip propagation delay calculated by dividing 27800 m by the speed of light. Rerunning the experiment with the increased slot size is shown in Figure 3.11 and illustrates that the channel is now shared between the two nodes. The drop off at around 30000 m is a result of the mobile node moving out of transmission range. Note that since the DIFS time is increased, this results in decreased throughput because the nodes will wait longer before transmitting. It can be seen in the first test with the default slot time that the aggregate throughput was around 5 Mb/s whereas the aggregrate throughput was around 2.5 Mb/s with the increased slot time. So, the benefits and drawbacks of increasing the slot time should be considered carefully for a given scenario.

3.1.5 Choosing Miscellaneous Optimal Values

Section 3.1.2 details how to configure ns-3 to work with longer transmission ranges, Section 3.1.3 shows that using the short PLCP preamble resulted in greater throughput, and Section 3.1.4 details how to configure the simulations such that the channel is shared fairly between nodes that are close and nodes that are further from the destination. With these enhancements, we now want to determine the optimal transmission power, data rate, and MAC level retransmission rate to run

the simulations with to test the performance of different routing protocols.

Even with the discussed enhancements, low packet delivery ratio (PDR) values were observed when increasing the node density such as with the star pattern tests. This is due to the ns-3 simulation of the MAC layer, which uses a Gaussian interference model coupled with a bit-error-rate (BER) model to determine whether or not a packet is received. This is different than the ns-2 approach to packet reception, which involves specifying a transmission power and a receiving power threshold. Since this research is focused on measuring the performance of the routing layer, lower sending rates were tested with in order to reduce interference amongst nodes as much as possible to get higher PDR values independent of a specific routing protocol.

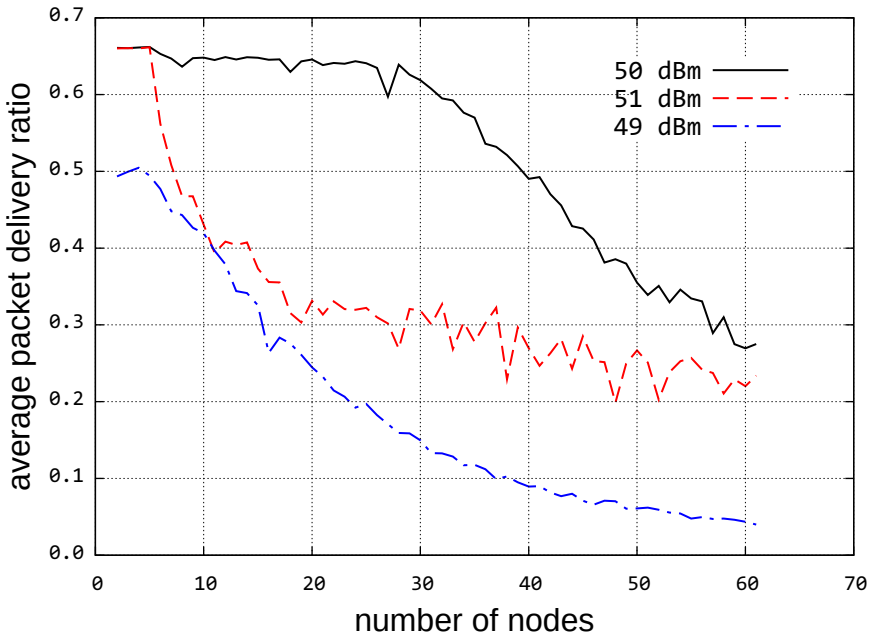

Figure 3.12. Various transmission powers in star pattern

The same star pattern test as discussed in Section 3.1.2 and Section 3.1.3 with the previous enhancements of the ACK and CTS timeout, short PLCP pream-

ble, and adjusted DIFS size were tested with nodes sending at 8 kb/s, 16 kb/s, and 24 kb/s rates and transmission powers of 49 dBm, 50 dBm, and 51 dBm. Section 3.1.2 showed that 51 dBm yielded the highest throughput but transmission ranges of 49 dBm and 50 dBm still provided connectivity at the 27800 m transmission range. Figure 3.12 illustrates the PDR for the various transmission powers when averaging the different sending rates of 8 kb/s, 16 kb/s, and 24 kb/s. A transmission power of 50 dBm was shown to have the best PDR in these tests.

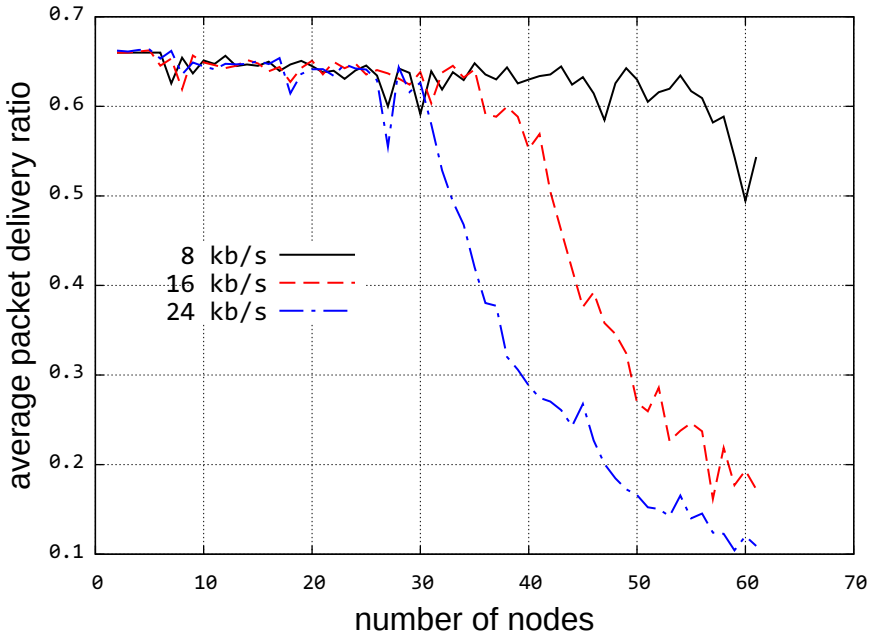

Figure 3.13. Various data rates in star pattern

Figure 3.13 shows the results of the same simulation but with a transmission power of 50 dBm and the individual sending rates of the nodes shown. The sending rate of 8 kb/s or 1 packet/s yielded the best PDR.

The final test was to test with the MAC level data retransmission rate. In 802.11, this is controlled by the station long retry count (SLRC) [49]. ns-3 controls this via the `MaxSLRC` parameter, which has a default of 7. Figure 3.14 illustrates

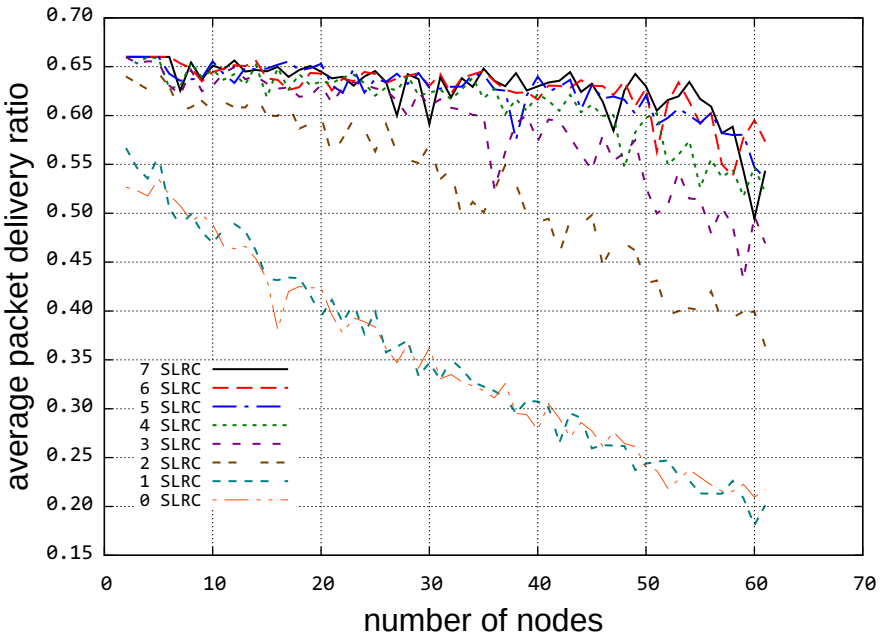

Figure 3.14. Various SLRC values in star pattern

the results of simulating the star pattern simulation with a transmission power of 50 dBm, nodes sending at a rate of 8 kb/s, and varying the `MaxSLRC` value between 0 and 7 with 0 being no MAC level data retransmissions and 7 being up to 7 MAC level data retransmissions. The results showed that there was no more PDR gain after a `MaxSLRC` value of 5.

3.1.6 Conclusions

The conclusions based on these basic non-mobile performance tests are:

1. Significant consideration should be taken when determining the transmission power because small changes can greatly effect maximum throughput.

2. Maximum throughput is negatively effected as the number of sending nodes in the network is increased.

3. The CTS and ACK timeout values need to be increased outside the ns-3 default when simulating long transmission ranges.

4. The DIFS slot time can be adjusted to allow for fair sharing of the channel between nodes that are close to the sink and those that are far away. However, the aggregate throughput will suffer as a result of increasing the DIFS.

5. Lower data rates, a slightly lower transmission power than the optimal power between a single sender and receiver, and a slightly adjusted MAC level data retransmission rate are desirable for routing protocol performance testing in ns-3.

3.2 AeroRP Decision Metrics

The following sections discuss the different metrics that AeroRP uses to make its forwarding decisions, maintenance of its neighbor list, and when to ferry a data packet.

3.2.1 Speed Component

The speed component (s_{d}) is used in the TTI calculation discussed in Section 3.2.2. s_{d} is the relative speed a potential neighbor has with respect to the destination. This gives the TTI both a speed and directional component that a neighbor has with regards to the destination. A high and positive s_{d} infers the neighbor is moving towards the destination at a high speed. A high and negative s_{d} infers the neighbor is moving away from the destination at a high speed. The assumption is that a node knows its own location, velocity, and the destination's

location. Given a neighbor n_i that has geographical coordinates of x_i, y_i and a velocity of v_{xi}, v_{yi}, the velocity for n_i is calculated as:

$$v_i = \sqrt{{v_{xi}}^2 + {v_{yi}}^2} \tag{3.2}$$

`atan2(x,y)` is a two-argument convenience function available in most programming languages that computes the angle in radians between the positive x-axis of a plane and the x, y coordinates provided in the arguments [50]. In terms of arctan, `atan2(x,y)` is short hand for:

$$\text{atan2}(y,x) = \begin{cases} \arctan(\frac{y}{x}) & x > 0 \\ \pi + \arctan(\frac{y}{x}) & y \geq 0, x < 0 \\ -\pi + \arctan(\frac{y}{x}) & y < 0, x < 0 \\ \frac{\pi}{2} & y > 0, x = 0 \\ -\frac{\pi}{2} & y < 0, x = 0 \\ \text{undefined} & y = 0, x = 0 \end{cases}$$

The angle in degrees between the positive x-axis of n_i plane and n_i velocity is given as follows:

$$\Theta = \text{atan2}(v_{yi}, v_{xi}) \times \frac{180}{\pi} \tag{3.3}$$

Destination D has geographical coordinates x_d, y_d. To calculate the angle between the positive x-axis of n_i plane and the imaginary line drawn between n_i and D is given as follows:

$$\bar{\Theta} = \text{atan2}(y_\text{d} - y_i, x_\text{d} - x_i) \times \frac{180}{\pi} \tag{3.4}$$

The difference between the angles $(\Theta - \bar{\Theta})$ gives the angle between n_i velocity and the imaginary line drawn between n_i and D. Taking the cosine of this angle and the product of n_i velocity (Equation 3.2) gives us s_d:

$$s_\mathrm{d} = v_i \times \cos(\Theta - \bar{\Theta}) \tag{3.5}$$

Figure 3.15 illustrates an example in which a potential neighbor is moving towards the destination in quadrant I relative to the destination. The s_d calculation is calculated out as follows:

$$\begin{aligned}
v_i &= \sqrt{{v_{xi}}^2 + {v_{yi}}^2} \\
&= \sqrt{-14.15^2 + -14.15^2} \\
&\approx 20 \text{ m/s} \\
\Theta &= \text{atan2}(v_{yi}, v_{xi}) \times \frac{180}{\pi} \\
&= \text{atan2}(-14.15, -14.15) \times \frac{180}{\pi} \\
&= -135.2^\circ \\
\bar{\Theta} &= \text{atan2}(y_\mathrm{d} - y_i, x_\mathrm{d} - x_i) \times \frac{180}{\pi} \\
&= \text{atan2}(200 - 1200, 600 - 1000) \times \frac{180}{\pi} \\
&= -111.7^\circ \\
s_\mathrm{d} &= v_i \times \cos(\Theta - \bar{\Theta}) \\
&= 20 \text{ m/s} \times \cos(-135.2^\circ - (-111.7^\circ)) \\
&= 18.4 \text{ m/s}
\end{aligned}$$

Figure 3.16 illustrates an example in which a potential neighbor is moving away

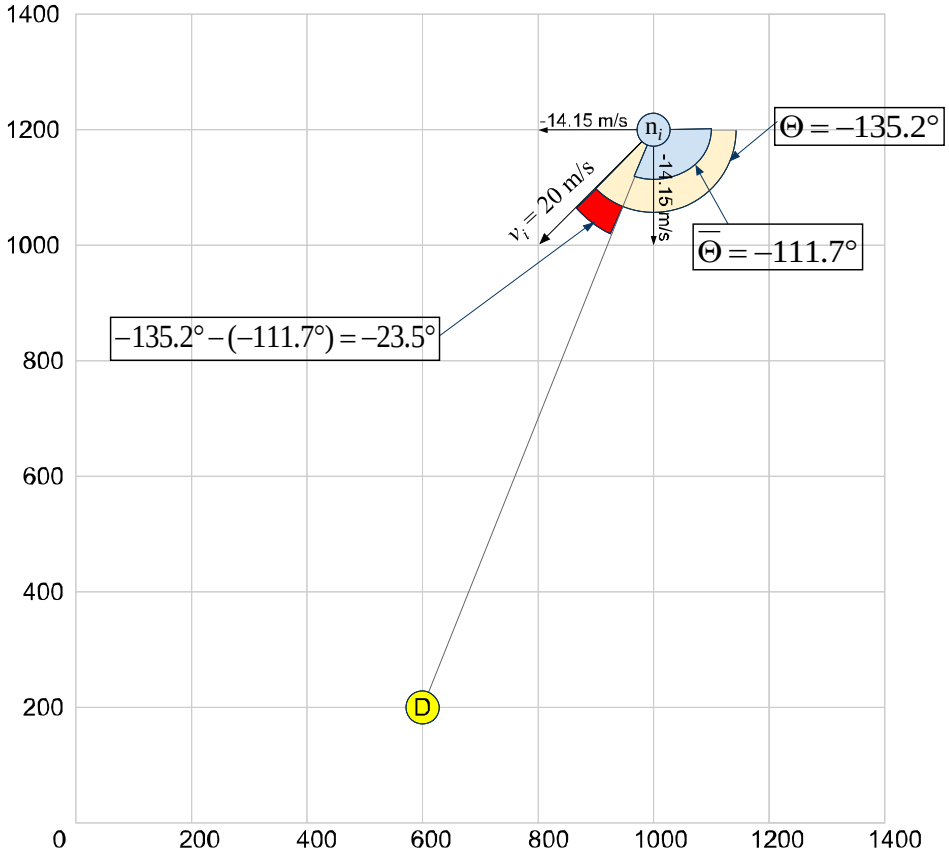

Figure 3.15. Sample s_{d} – towards destination

from the destination in quadrant I relative to the destination. The s_{d} calculation is calculated as follows:

$$
\begin{aligned}
v_i &= \sqrt{{v_{xi}}^2 + {v_{yi}}^2} \\
&= \sqrt{14.15^2 + 14.15^2} \\
&\approx 20 \text{ m/s} \\
\Theta &= \operatorname{atan2}(v_{yi}, v_{xi}) \times \frac{180}{\pi} \\
&= \operatorname{atan2}(14.15, 14.15) \times \frac{180}{\pi} \\
&= 44.69°
\end{aligned}
$$

$$
\begin{aligned}
\bar{\Theta} &= \text{atan2}(y_\text{d} - y_i, x_\text{d} - x_i) \times \frac{180}{\pi} \\
&= \text{atan2}(200 - 1200, 600 - 1000) \times \frac{180}{\pi} \\
&= -111.7^\circ \\
s_\text{d} &= v_i \times \cos(\Theta - \bar{\Theta}) \\
&= 20 \text{ m/s} \times \cos(44.69^\circ - (-111.7^\circ)) \\
&= -18.4 \text{ m/s}
\end{aligned}
$$

Figure 3.17 illustrates an example in which a potential neighbor is moving away

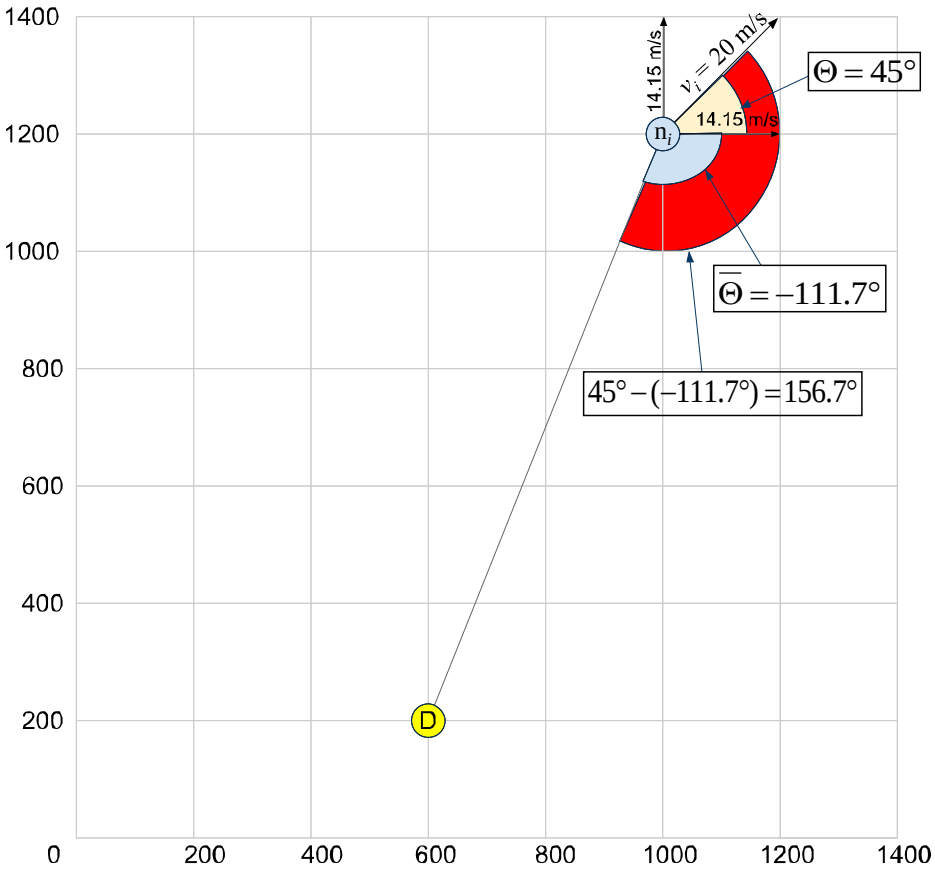

Figure 3.16. Sample s_d – away from destination

from the destination on the x-axis and towards the destination on the y-axis in

quadrant II relative to the destination. The s_{d} calculation is calculated as follows:

$$
\begin{aligned}
v_i &= \sqrt{{v_{xi}}^2 + {v_{yi}}^2} \\
&= \sqrt{-14.15^2 + 14.15^2} \\
&\approx 20 \text{ m/s} \\
\Theta &= \operatorname{atan2}(v_{yi}, v_{xi}) \times \frac{180}{\pi} \\
&= \operatorname{atan2}(14.15, -14.15) \times \frac{180}{\pi} \\
&= 135.22^\circ \\
\bar{\Theta} &= \operatorname{atan2}(y_{\mathrm{d}} - y_i, x_{\mathrm{d}} - x_i) \times \frac{180}{\pi} \\
&= \operatorname{atan2}(1200 - 200, 1000 - 600) \times \frac{180}{\pi} \\
&= 68.18^\circ \\
s_{\mathrm{d}} &= v_i \times \cos(\Theta - \bar{\Theta}) \\
&= 20 \text{ m/s} \times \cos(135.22^\circ - 68.18^\circ) \\
&= 7.8 \text{ m/s}
\end{aligned}
$$

Figure 3.18 illustrates an example in which a potential neighbor is moving only on the x-axis from the north directly towards the destination. The s_{d} calculation is calculated as follows:

$$
\begin{aligned}
v_i &= \sqrt{{v_{xi}}^2 + {v_{yi}}^2} \\
&= \sqrt{20^2 + 0^2} \\
&\approx 20 \text{ m/s} \\
\Theta &= \operatorname{atan2}(v_{yi}, v_{xi}) \times \frac{180}{\pi}
\end{aligned}
$$

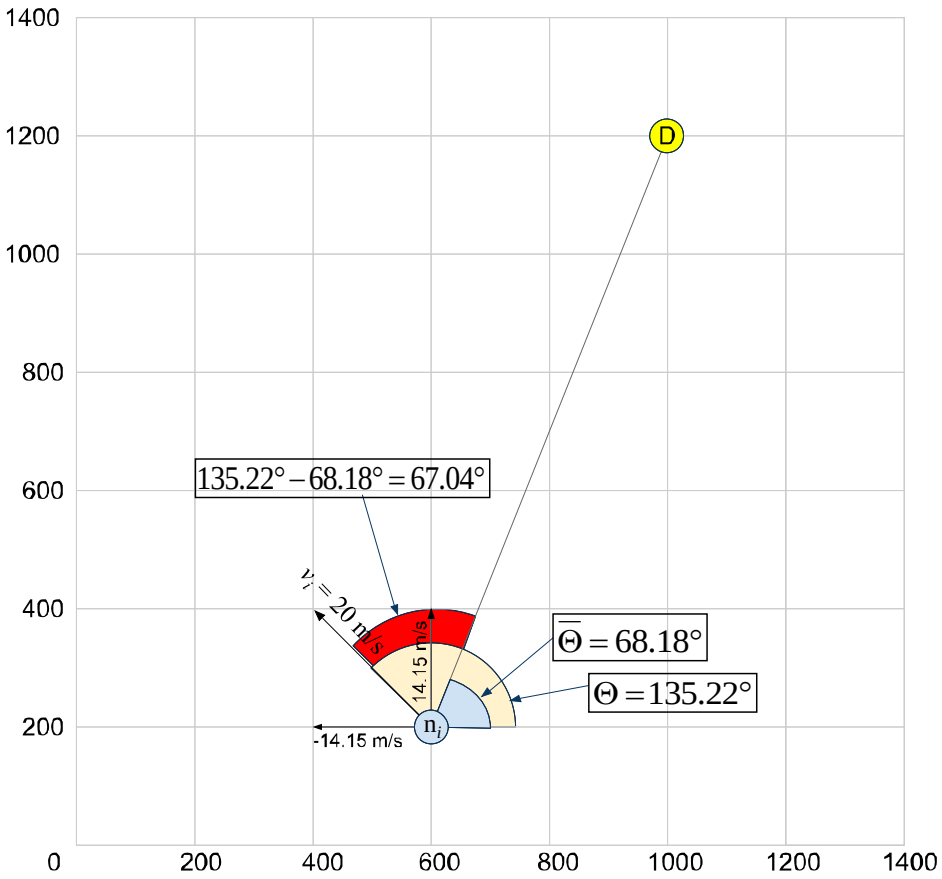

Figure 3.17. Sample s_{d} – towards & away from destination

$$
\begin{aligned}
&= \operatorname{atan2}(0, 20) \times \frac{180}{\pi} \\
&= 0^\circ \\
\bar{\Theta} &= \operatorname{atan2}(y_{\mathrm{d}} - y_i, x_{\mathrm{d}} - x_i) \times \frac{180}{\pi} \\
&= \operatorname{atan2}(800 - 800, 1000 - 400) \times \frac{180}{\pi} \\
&= 0^\circ \\
s_{\mathrm{d}} &= v_i \times \cos(\Theta - \bar{\Theta}) \\
&= 20 \text{ m/s} \times \cos(0^\circ - 0^\circ) \\
&= 20 \text{ m/s}
\end{aligned}
$$

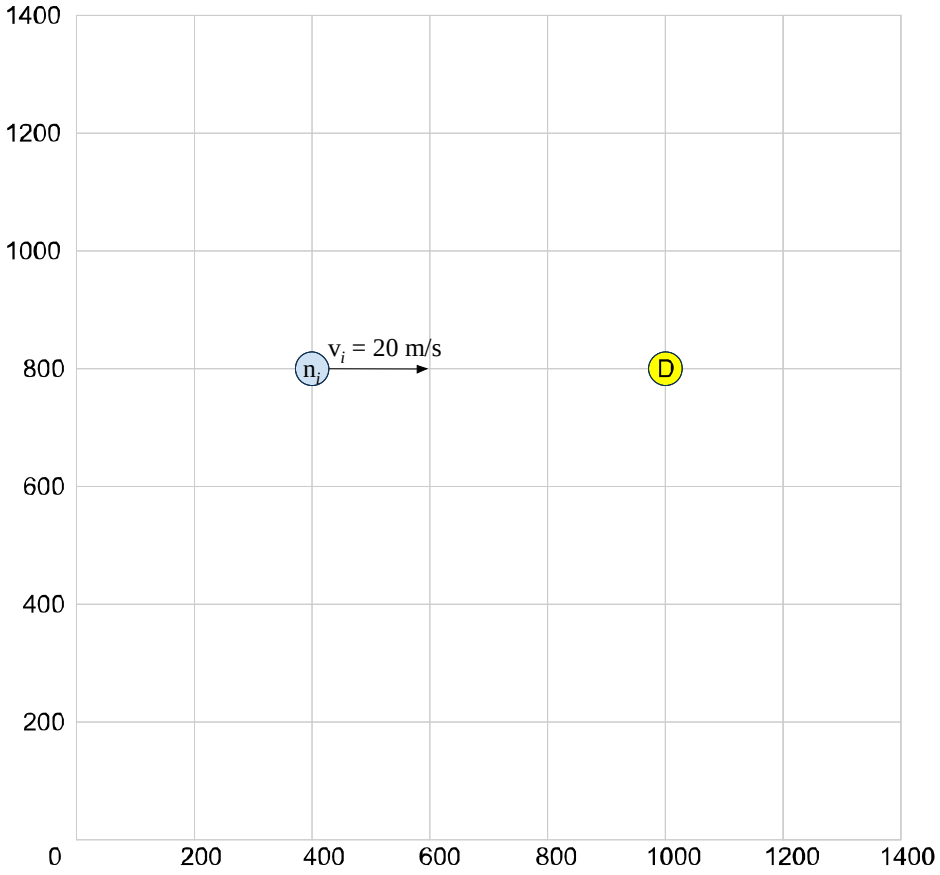

Figure 3.18. Sample s_{d} – direct from North

3.2.2 Time to Intercept

The *time to intercept* (TTI) is the primary metric used for routing decisions in AeroRP. A source node calculates the TTI of its neighbors to understand when its neighbors will potentially be within transmission range of the destination and make the decision to route to the neighbor that will potentially be within transmission range the soonest and thus has the lowest TTI.

Given a potential neighbor n_i with coordinates x_i, y_i, z_i and a destination D

with coordinates $x_{\mathrm{d}}, y_{\mathrm{d}}, z_{\mathrm{d}}$, the Euclidean distance between the two is given as:

$$\Delta d = \sqrt{(x_{\mathrm{d}} - x_i)^2 + (y_{\mathrm{d}} - y_i)^2 + (z_{\mathrm{d}} - z_i)^2} \tag{3.6}$$

The TTI is calculated as follows in which R is the transmission range of the mobile devices and the s_{d} is calculated from Section 3.2.1, Equation 3.5:

$$\mathrm{TTI} = \begin{cases} 0 & \text{for } s_{\mathrm{d}} < 0 \text{ and } \Delta d > R \\ \frac{\Delta d - R}{s_{\mathrm{d}}} & \text{otherwise} \end{cases} \tag{3.7}$$

TTI=0 is a special case that indicates to never choose a neighbor when its TTI is 0 as a next hop because we do not want to choose nodes 1) that are moving away from the destination and 2) not within transmission range of the destination. Figure 3.19 illustrates the problem that can occur when taking this case into account. If the TTI of the neighbors (n_1 and n_2) is just calculated as $\frac{\Delta d - R}{s_{\mathrm{d}}}$, the n_1 node will be chosen as the next hop by n_0 because it will have a negative TTI. However, n_2 is the best choice because it is moving towards the destination. To fix this, the piecewise version of the TTI calculation should be used such that the TTI of n_1 would be calculated as 0 and AeroRP would know to never choose this node as a next hop and the better choice of n_2 would be the next hop.

We still want nodes to have a negative TTI because this is an indication of a node being within transmission range of the destination and should be chosen as a next hop (these nodes will have the lowest TTI). We always want nodes within transmission range of the destination to be chosen over nodes that are not. We also want nodes that are within transmission range of the destination but moving away from the destination to be chosen if there are no nodes within transmission

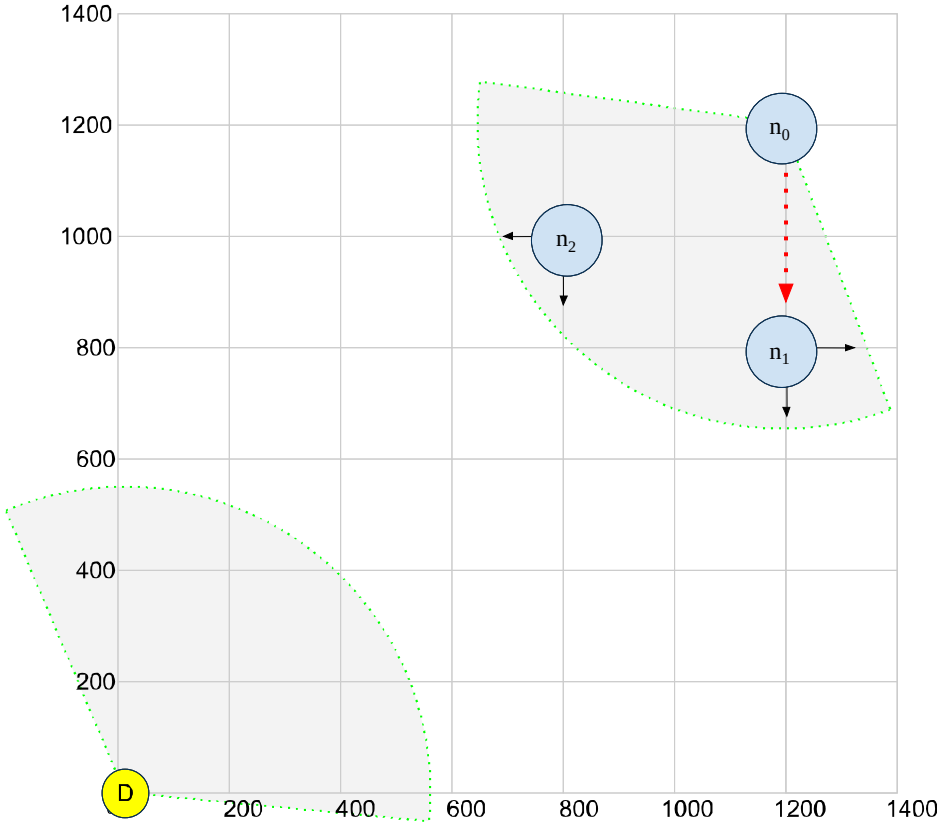

Figure 3.19. Consideration for negative TTI

range of the destination that are moving towards the destination. The nodes within transmission range moving towards the destination will be favored over those nodes within transmission range but moving away from the destination because these nodes will have a positive TTI due to a negative s_{d} and a negative $\Delta d - R$, and the nodes moving towards the destination will have a negative TTI.

3.2.3 Expiring Neighbors

There are two ways that AeroRP removes nodes from its neighbor list that it has collected geographic information about:

1. purging stale entries

2. predict neighbors that will be out of range

Purging stale entries is straightforward; a configurable variable is passed to AeroRP that indicates how long to hold entries in the neighbor table. During neighbor table housekeeping and routing decision operations, neighbors that exceed this hold time will be purged from the neighbor list and are thus removed as a next hop consideration.

In a highly dynamic mobile environment in which links are constantly being broken due to high speeds, it may not be enough to just purge entries that have not been heard from based on the configurable hold time. Hence, we try to predict nodes that will be out of transmission range and remove them from next hop consideration. Given the following:

$$\begin{aligned}
\text{Current source } (\mathrm{n}_0) \text{ location} &: x_0, y_0, z_0 \\
\text{Last known location of potential next hop } (\mathrm{n}_i) &: x_i, y_i, z_i \\
\text{Last known velocity of potential next hop} &: v_{xi}, v_{yi}, v_{zi} \\
\text{Transmission range} &: R \\
\text{Time last } \mathrm{n}_i \text{ info was received} &: t_0 \\
\text{Current time} &: t_1
\end{aligned}$$

The predicted distance ($\hat{d}$) between n_0 and n_1 is:

$$\begin{aligned}
\bar{x}_i &= x_i + v_{xi}(t_1 - t_0) \\
\bar{y}_i &= y_i + v_{yi}(t_1 - t_0) \\
\bar{z}_i &= z_i + v_{zi}(t_1 - t_0) \\
\hat{d} &= \sqrt{(x_0 - \bar{x}_i)^2 + (y_0 - \bar{y}_i)^2 + (z_0 - \bar{z}_i)^2}
\end{aligned} \tag{3.8}$$

Finally, the logic used to predict whether or not n_i is going to be out of n_0 range is given as:

$$\text{OutOfRange} = \begin{cases} \text{true} & \text{for } \hat{d} \geq R \\ \text{false} & \text{for } \hat{d} < R \end{cases} \tag{3.9}$$

Figure 3.20 shows n_0 and n_1 moving and still being within transmission range of

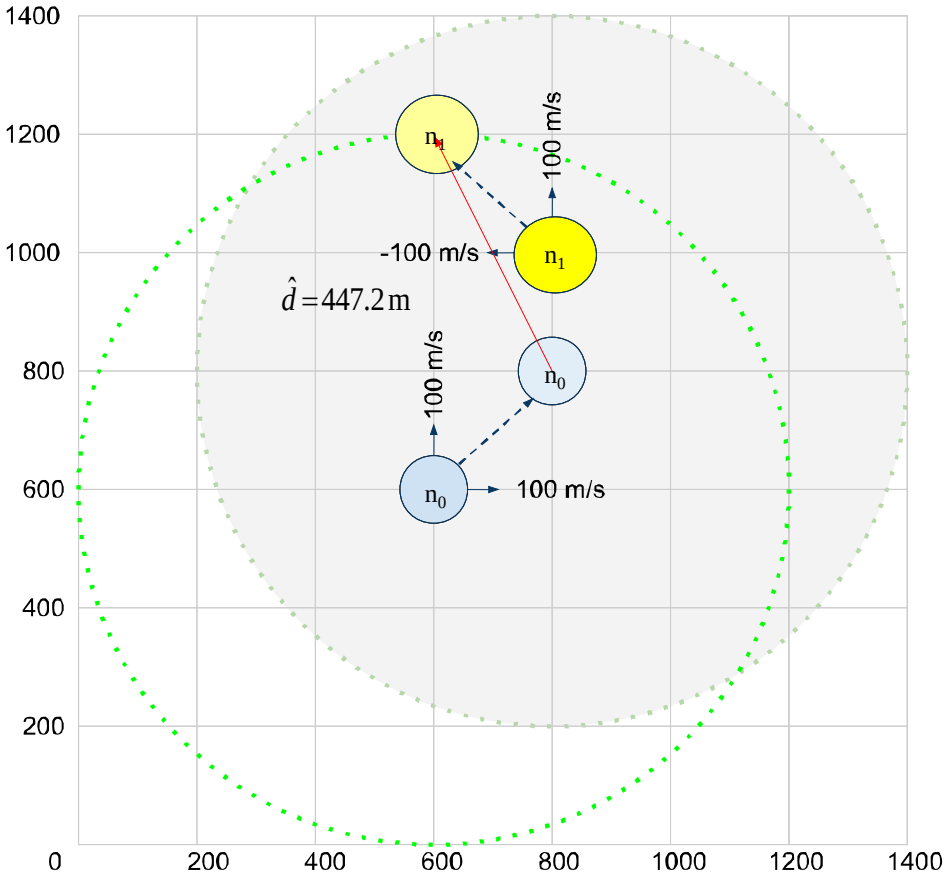

Figure 3.20. Nodes moving and staying within range

each other. The transmission range R is 600 m, $t_0 = 60s$, and $t_1 = 62s$, calculated as follows:

$$\bar{x}_i = x_i + v_{xi}(t_1 - t_0)$$

$$
\begin{aligned}
&= 800 + \ -100(62 - 60) \\
&= 600 \\
\bar{y}_i &= y_i + v_{yi}(t_1 - t_0) \\
&= 1000 + 100(62 - 60) \\
&= 1200 \\
\bar{z}_i &= z_i + v_{zi}(t_1 - t_0) \\
&= 0 + 0(62 - 60) \\
&= 0 \\
\hat{d} &= \sqrt{(x_0 - \bar{x}_i)^2 + (y_0 - \bar{y}_i)^2 + (z_0 - \bar{z}_i)^2} \\
&= \sqrt{(800 - 600)^2 + (800 - 1200)^2 + (0 - 0)^2} \\
&= 447.2 \text{ m} \\
\text{OutOfRange} &= \begin{cases} \text{true} & \text{for } \hat{d} \geq R \\ \text{false} & \text{for } \hat{d} < R \end{cases} \\
&= \text{false}
\end{aligned}
$$

Figure 3.21 shows n_0 and n_1 moving out of transmission range from each other. The transmission range (R) is 600 m, $t_0 = 60s$, and $t_1 = 62s$. The calculation is as follows:

$$
\begin{aligned}
\bar{x}_i &= x_i + v_{xi}(t_1 - t_0) \\
&= 800 + \ -200(62 - 60) \\
&= 400 \\
\bar{y}_i &= y_i + v_{yi}(t_1 - t_0)
\end{aligned}
$$

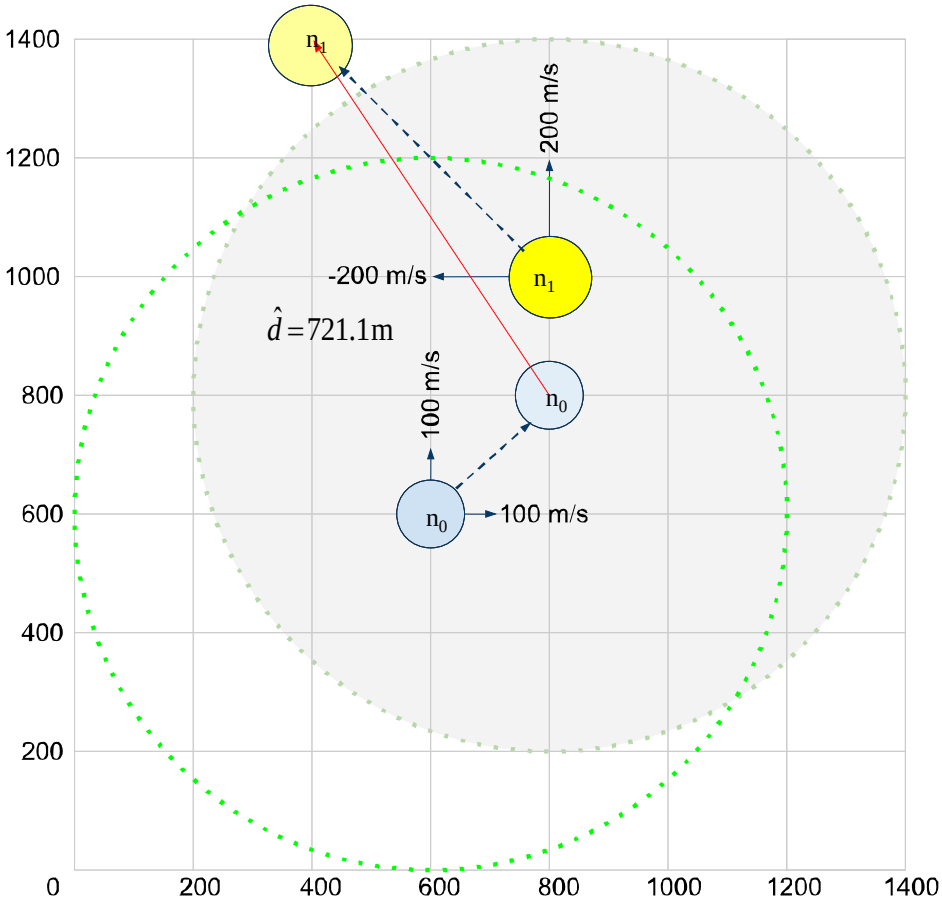

Figure 3.21. Nodes moving out of transmission range

$$= 1000 + 200(62 - 60)$$

$$= 1400$$

$$\bar{z}_i = z_i + v_{zi}(t_1 - t_0)$$

$$= 0 + 0(62 - 60)$$

$$= 0$$

$$\hat{d} = \sqrt{(x_0 - \bar{x}_i)^2 + (y_0 - \bar{y}_i)^2 + (z_0 - \bar{z}_i)^2}$$

$$= \sqrt{(800 - 400)^2 + (800 - 1400)^2 + (0 - 0)^2}$$

$$= 721.2 \text{ m}$$

$$\text{OutOfRange} = \begin{cases} \text{true} & \text{for } \hat{d} \geq R \\ \text{false} & \text{for } \hat{d} < R \end{cases}$$

$$= \text{true}$$

3.2.4 Ferrying

For the case when the node receives a packet for which the node itself has the best TTI but is not within transmission range of the destination, the packet is queued in a configurable sized queue for a configurable amount of time. The queue is checked at a configurable frequency to see if there is a neighbor with a lower TTI than the ferrying node. When a neighbor with a lower TTI is encountered, the

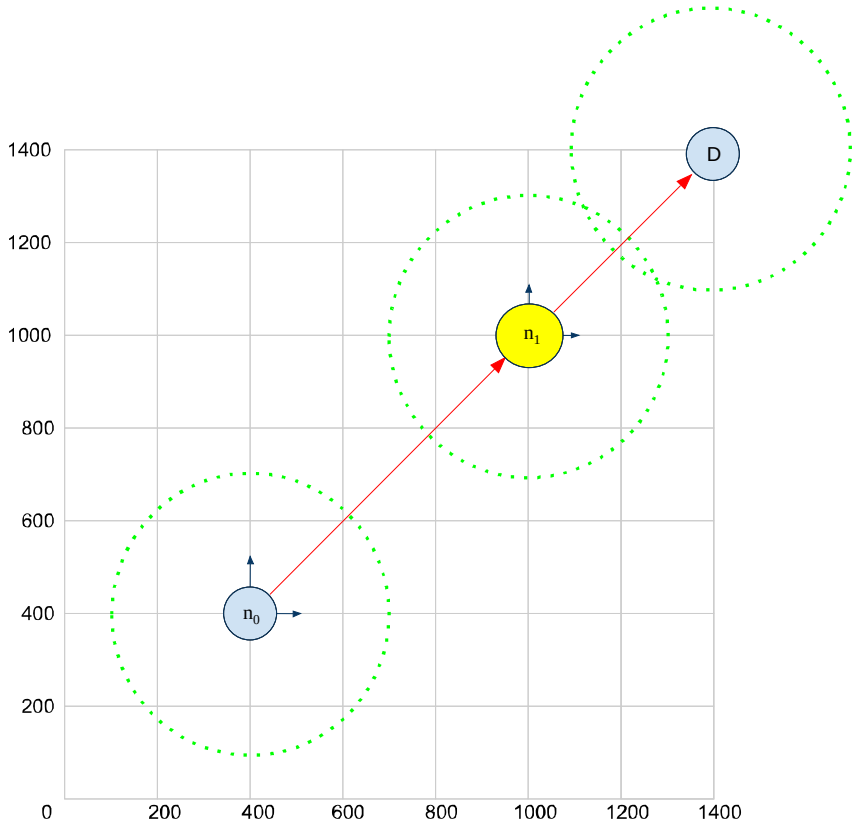

Figure 3.22. Ferrying packet until better TTI is encountered

packets from the queue are sent at the configurable ferry data rate. Figure 3.22 illustrates the need for n_0 to queue its packets and ferry the packets until within transmission range of n_1, which has a better TTI than n_0.

3.3 AeroRP in ns-3

This section details how AeroRP was implemented in ns-3.

3.3.1 Implementation Flow

The flow of the AeroRP implementation can be separated into three separate activities:

1. send `hello` beacon
2. receive `hello` beacon
3. receive data packet

Sending a `hello` beacon is straightforward and is illustrated in Figure 3.23. A node simply constructs an AeroRP `hello` beacon packet consisting of its location and velocity details and broadcasts it (packet format detailed in Section 3.3.2).

Nodes receive AeroRP `hello` beacons from their neighbors by listening on a predetermined port number. The node either creates a new entry or updates its current data regarding the node it received the `hello` beacon from. At this time, the node will also take this opportunity to check its current neighbor list for any nodes that have not been heard from for a predetermined amount of time and purge them from its neighbor list. Expiring neighbors is discussed in more detail in Section 3.2.3. The flow of receiving `hello` beacons is illustrated in Figure 3.24.

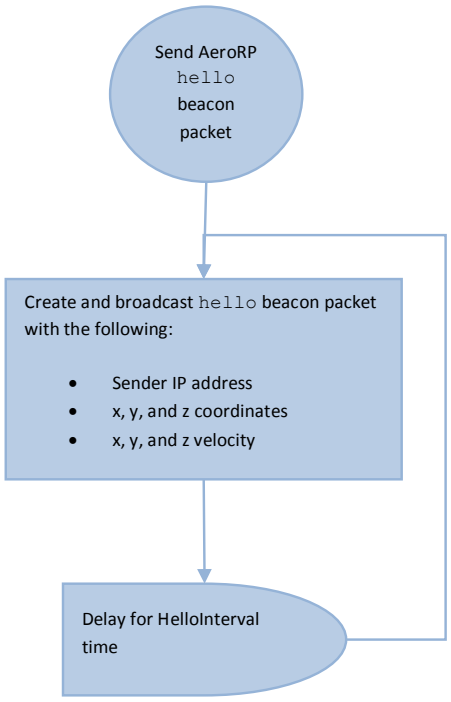

Figure 3.23. AeroRP – send `hello` beacon

When a node receives a data packet, it has to determine how to handle it. The following basic outline for handling data packets is illustrated in Figure 3.25:

1. if the packet has reached its destination then move the packet up the network layers for local delivery

2. purge expired nodes from the neighbor list (discussed in detail in Section 3.2.3)

3. if one of the neighbors is the destination of the packet, then send the packet to the destination

4. calculate the TTI (discussed in detail in Section 3.2.2) for all neighbors and choose the node with the lowest non-zero TTI

    - if the best TTI is from the local node, then queue the packet for ferrying

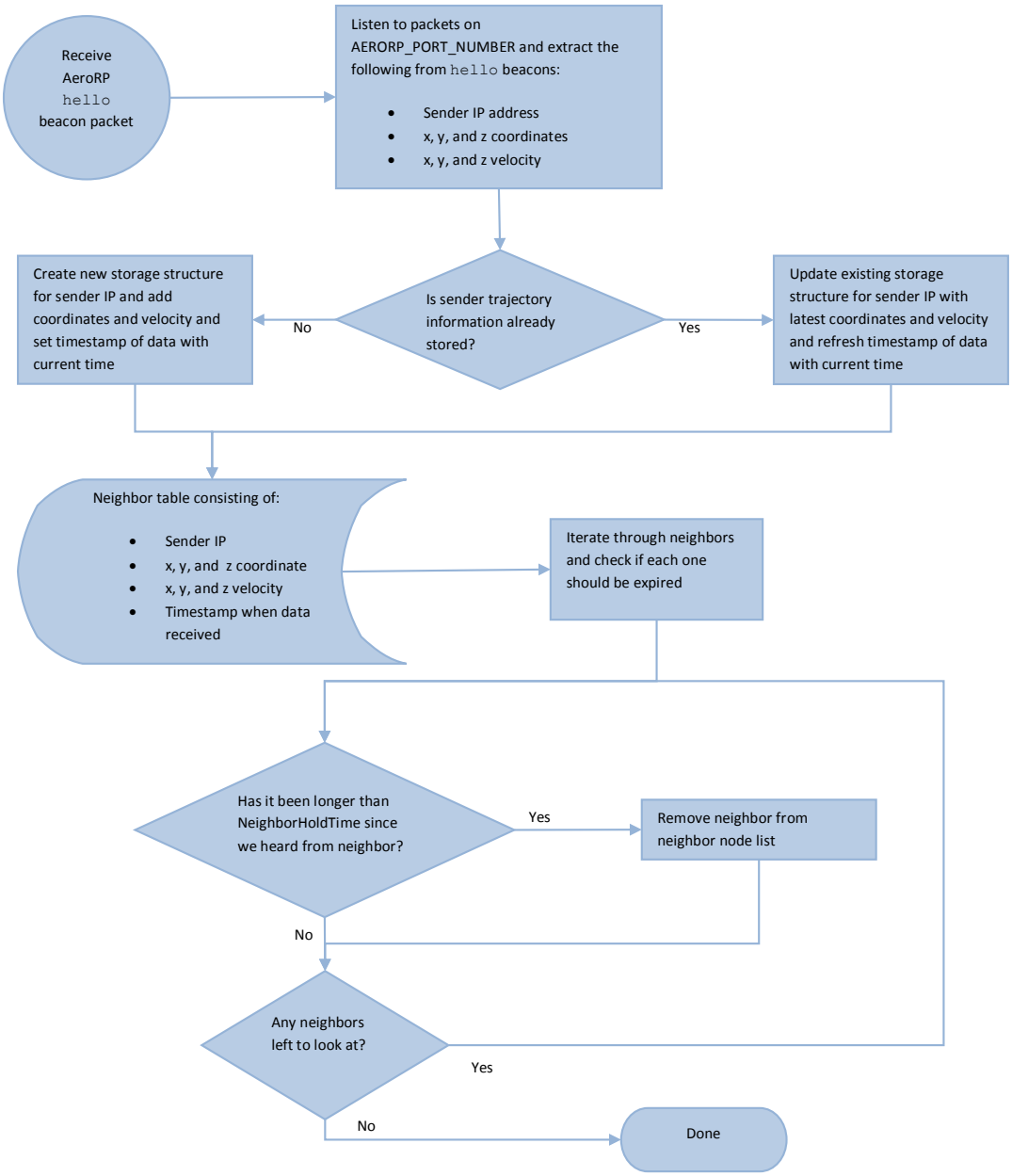

Figure 3.24. AeroRP – receive `hello` beacon

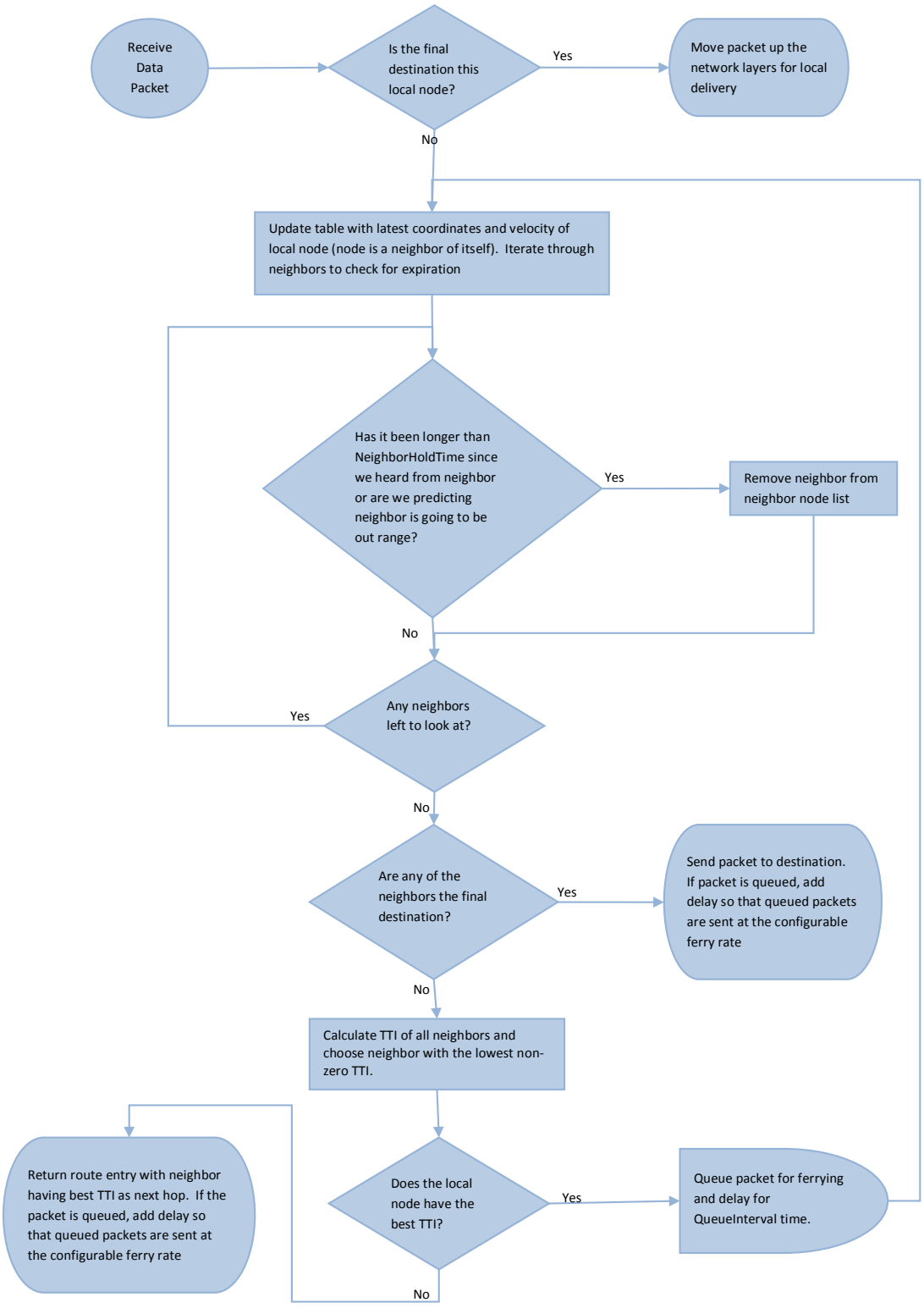

Figure 3.25. AeroRP – receive data packet

(if enabled) and check the queue later; if ferrying is not enabled, drop the packet

- if the best TTI is not from the local node, then send the packet to the next hop, which is the neighbor with the best TTI

3.3.1.1 Beaconless Promiscuous Mode

Given the wireless nature of node communication in MANETs, it is possible for a node to be promiscuous and overhear all packets, even those packets that are not intended for a given node. The beaconless promiscuous mode of AeroRP takes advantage of this behavior and adds location information to each data packet per-hop as opposed to sending periodic `hello` beacons with this information. All nodes within transmission range, including those nodes that are not the intended receiver, can listen to the data packet and extract the location information from the header and store this location information for making routing decisions.

The TTI calculation, decisions for expiring neighbors, and ferrying remain basically the same in this mode with a few subtle differences. Since, in this mode, the location information is extracted from the data packets and no data packets can be sent to other nodes without location information, there has to be some way to start the flow of data. To do this, all nodes send one bootstrap `hello` beacon (Section 3.3.2) at a configurable time. The `hello` beacon is sent the same as in beacon mode per Figure 3.23.

The nodes operate in promiscuous mode to overhear all nearby traffic. Figure 3.26 shows the flow for when a node receives a data packet that is not intended for that node. The flow is exactly the same as when a node hears a `hello` beacon (Figure 3.24) except for extracting the location information from the header of

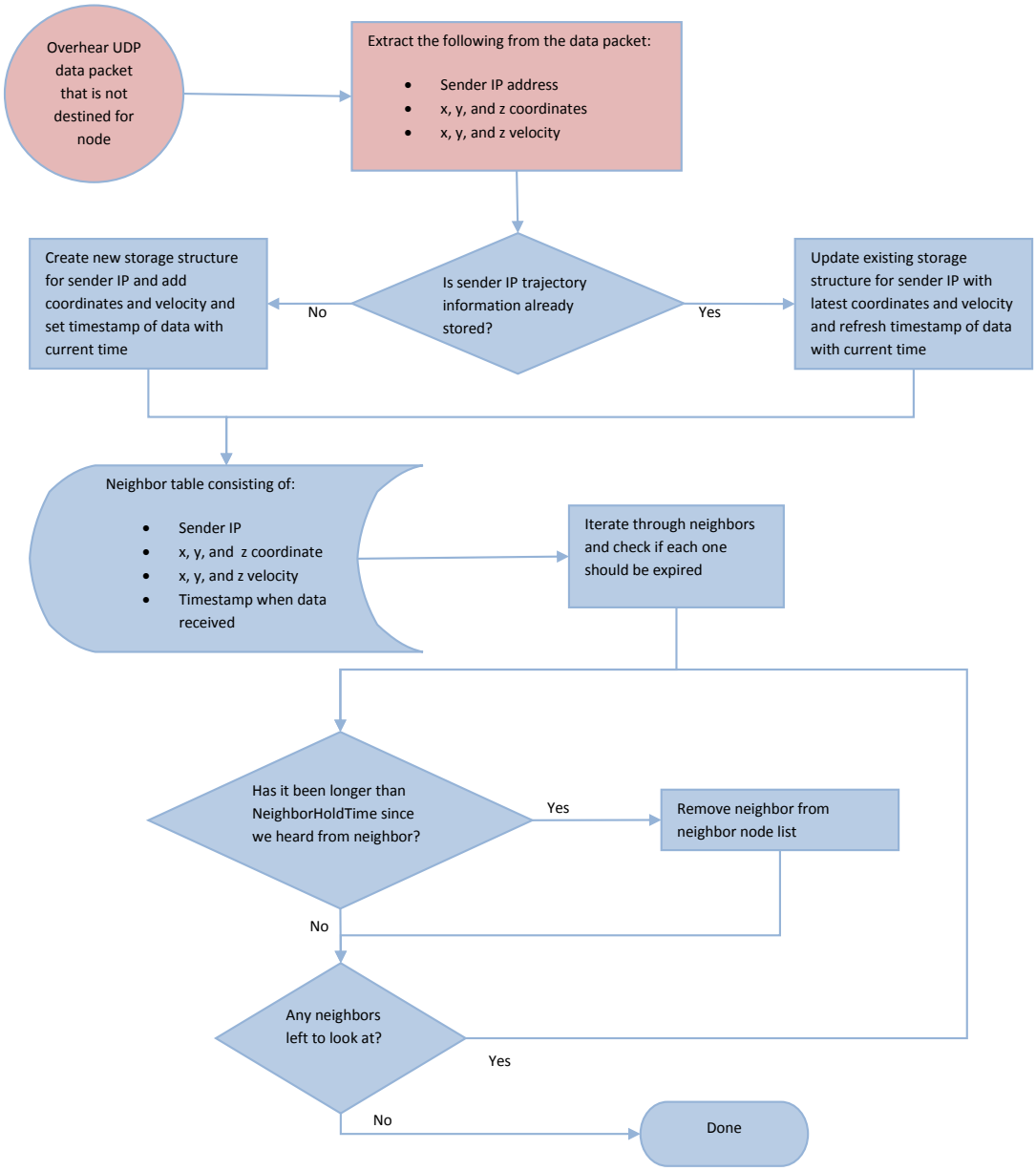

Figure 3.26. AeroRP beaconless – receive promiscuous data packet

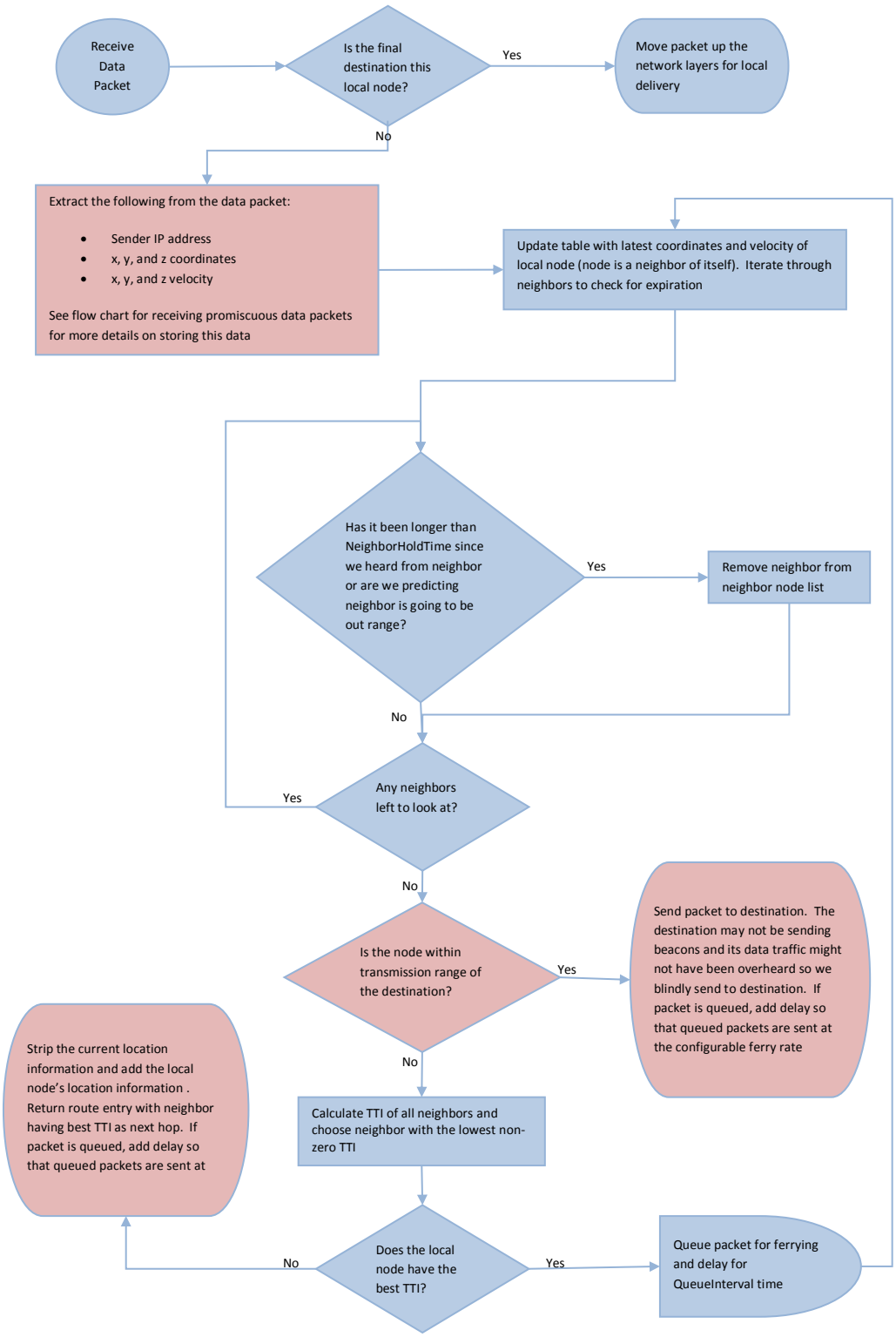

Figure 3.27. AeroRP beaconless – receive data packet for routing

actual data packets.

There are a few differences in beaconless promiscuous mode when receiving a packet intended for the node as illustrated by Figure 3.27. When the packet is received and intended for the node, the same process of extracting the location information as with an unindented recipient is carried out (Figure 3.26). One key difference in beaconless mode is the fact that the destination will not be sending beacons, so the destination will not be in the list of a node's location data unless the destination itself is also sending data. To remedy this, since all nodes already know the location of the final destination, the node will blindly send the data packet to the destination if it knows that it is within transmission range even if it has not overheard any packets from the destination. If the destination is not within transmission range and the local node has the best TTI, the packet can still have the option to be ferried. If the destination is not within transmission range and there are other nearby nodes that have a better TTI, the location information from the data packet is stripped off, the local node's location information is added, and then the packet is sent out to the neighbor with the better TTI.

3.3.2 `Hello` Beacon Packet Format

The `hello` beacon packet and the extra header written to data packets in beaconless promisucous mode conform with the ongoing development of the AeroNP specification [51, 52]. AeroNP is a network protocol for the highly-dynamic airborne environment. Figure 3.28 illustrates the current fields of the AeroNP header that AeroRP currently uses to construct the `hello` beacons and the header written to data packets in beaconless promisucous mode. The sections marked `Reserved` are not currently used in the AeroRP implementation. Since, as of this writing,

AeroNP is not implemented in ns-3, IP is used for the simulation network protocol. To account for the fact that an IP header is 20 bytes and the AeroNP header contains redundant data that the IP header already has, the last 20 `Reserved` bytes are not written out in the actual implementation so that protocol overhead is fairly counted for both `hello` beacons as well as the extra header written to data packets in beaconless promiscuous mode.

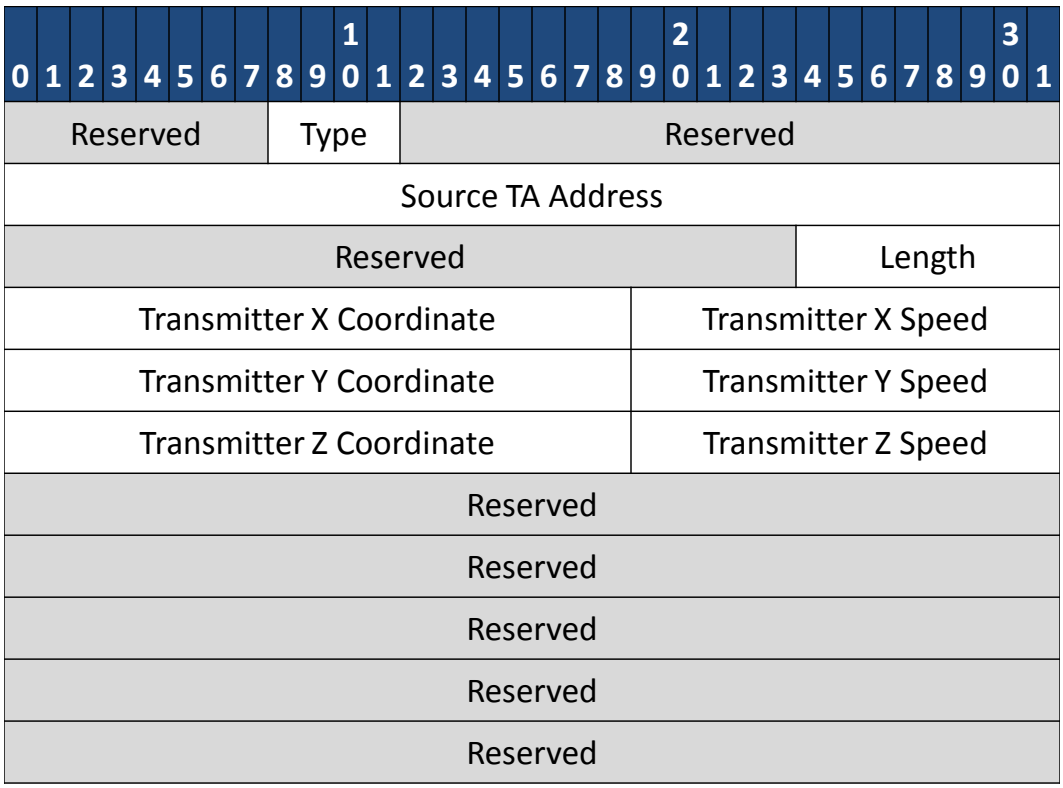

Figure 3.28. `hello` beacon packet format

3.4 Configuration

The simulation scripts were written to test the effect of multiple variables with respect to AeroRP and other routing protocols in a highly dynamic network:

1. mobility model
2. velocity
3. node density
4. AeroRP modes

The simulation randomly distributes a variable number of nodes throughout the simulation area. All of the nodes are configured to send constant bit rate (CBR) user datagram protocol (UDP) traffic at 8 kb/s and are all sending to the same location which is a sink node placed exactly in the middle of the simulation area. The sink node is meant to simulate the ground station that is receiving telemetry data from airborne nodes.

The simulation runs with the various permutations of parameters discussed in Section 3.4.1 and Section 3.4.2, and dumps all of the data to a single CSV (comma separated values) file. Each second of the simulation is written out to the CSV file as a single row that indicates how many packets are received at the sink node for that second, what routing protocol is being used, what mobility model is being used, how many nodes are in the simulation, and other specific details. This allows for detailed analysis and comparison of how changes in the simulation perform.

For overall analysis, post processing of the ns-3 trace files is conducted to calculate average packet delivery ratio (PDR), average packet delay, average control overhead, and the total number of packets transmitted and received at the MAC layer. The control overhead includes the size of the IP and UDP headers of data packets as well as the total size of individual control packets that the various routing protocols use.

The following sections go into the details of the simulation parameters and their values as well as the mobility models used to generate the results discussed in Chapter 4.

3.4.1 Parameters

Table 3.2 details the variables of the simulations that are changed in order to observe how these changes effect the performance of the network. Different routing protocols available in ns-3 are tested as well as the AeroRP routing protocol that was developed as part of this research. The different modes that AeroRP can operate in are also simulated. AeroRP is tested with both a ferrying mode as discussed in Section 3.2.4 as well as a buffered mode. In ferry mode, the packet queues are configured such that they are so large and never timeout so that a packet delivered early in the simulation has the potential to be delivered at the very end of the simulation. In buffered mode, the packet queues are configured to be more finite and have a timeout to be more in line with the buffering that AODV and DSDV implements. Various node densities and velocities are also tested. Two different mobility models, detailed in Section 3.4.2, are tested. All permutations of these variables are simulated.

Table 3.2. Simulation variables

Variable	Values
Routing protocol	OLSR, AODV, DSDV, and AeroRP
AeroRP modes	Dropping packets with no route, buffering packets, and ferrying packets as well as with and without beaconless promiscuous mode enabled
Node density	10, 20, 30, 40, 50, 60, 70, 80, 90, and 100 nodes
Mobility model	Random waypoint and Gauss-Markov
Velocity	1 m/s, 10 m/s, 100 m/s, 200 m/s, 400 m/s, 600 m/s, 800 m/s, 1000 m/s, 1200 m/s, and a uniform distribution between 200 m/s and 1200 m/s

Table 3.3. General simulation parameters

Parameter	Value
Number of times to run each simulation	10
Simulation area	150000 m × 150000 m × 1000 m
Initial position allocator	Random rectangle
Warmup time	100 s
Application sending time	1000 s
Link layer	wifib-11mbs
Packet size	1000 bytes
Sending rate	8 kb/s
RTS/CTS?	No
Packet fragmentation?	No
Propagation loss model	Friis
Transmission power	50 dBm
Transport protocol	UDP

Table 3.3 lists the general ns-3 simulation parameters that do not necessarily fit in a specific category. The warmup time is set to 100 s in order to allow time for the mobility models to stabilize before data transmission begins. The sending rate and transmission power of 50 dBm is chosen based on the research done in Section 3.1 in which these were found to be the optimal values for transmission ranges up to 27800 m.

In addition to these general parameters, several MAC enhancements outside of the ns-3 default are implemented:

1. The ACK and CTS timeout is adjusted to account for the round trip propagation time between nodes that are 27800 m apart. This is discussed in detail in Section 3.1.2.

2. The PLCP preamble is changed from long to short in order to gain slightly better performance as discussed in detail in Section 3.1.3.

3. The patch from `http://www.nsnam.org/bugzilla/show_bug.cgi?id=737`

is applied as it is not in the ns-3.7.1 version of the simulator being used for this research. This patch is related to DCF and makes the backoff more random for medium access in order to avoid collisions.

4. The DIFS value is increased to account for the round trip propagation time between nodes that are 27800 m apart in order to allow for fair sharing of the channel. This is discussed in detail in Section 3.1.4.

5. The default of up to 7 data retransmission is decreased to up to 5 data retransmissions. Section 3.1.5 details the reasoning behind this change.

Table 3.4. OLSR parameters

Parameter	**Value**
HelloInterval	1 s
TcInterval	5 s
MidInterval	5 s

The few parameters that are configurable for OLSR are listed in Table 3.4. These are all the default settings with the exception of the HelloInterval being changed from 2 s to 1 s to be in line with the settings of the other routing protocols tested.

Table 3.5 lists the parameters used to configure the AODV routing protocol. A few of these are outside of the default. The RreqRateLimit, which controls how many RREQ's are transmitted per second was reduced from 10 to 5 to reduce flooding since the topology is rapidly changing. The DeletePeriod, which controls how quickly routes are marked as invalid, is reduced from 15 s to 8 s so that routes in a rapidly changing topology will be marked invalid more quickly. The NetDiameter is adjusted based on how many nodes are in a specific simulation; this value is used to help AODV estimate how long it should be waiting for replies

Table 3.5. AODV parameters

Parameter	Value
Hello interval	1 s
RreqRetries	2 retries for a route
RreqRateLimit	5 RREQ per second
NodeTraversalTime	40 ms
NextHopWait	50 ms
ActiveRouteTimeout	3 s
MyRouteTimeout	11.2 s
BlackListTimeout	5.6 s
DeletePeriod	8 s
NetDiameter	Number of nodes - 1
NetTraversalTime	2.8 s
PathDiscoveryTime	5.6 s
MaxQueueLen	64 packets
MaxQueueTime	30 s
AllowedHelloLoss	2 hellos
GratuitousReply	TRUE
DestinationOnly	FALSE
EnableHello	TRUE
EnableBroadcast	TRUE

from control packets.

Table 3.6. DSDV parameters

Parameter	Value
ForwardingInterval	4 s
SettlingTime	0 s
MaxQueueLen	Number of nodes × MaxQueuedPacketsPerDst
MaxQueuedPacketsPerDst	5 packets
MaxQueueTime	30 s
EnableBuffering	TRUE
EnableWST	FALSE
Holdtimes	3 × ForwardingInterval
EnableRouteAggregation	FALSE

DSDV in ns-3 was implemented by the ResiliNets group at the University of Kansas [53]. The parameters and their values used for testing DSDV are listed in Table 3.6. A few of these parameters have been changed to accommodate the

high mobility being tested. The ForwardingInterval is reduced from 8 s to 4 s so that the routing data is forwarded more often. The default for the SettlingTime is 3 s, but it has been reduced to 0 s so that DSDV never waits before considering a route valid since a route may not be valid for very long at such high speeds.

Table 3.7. AeroRP parameters

Parameter	Value
`hello` beacon interval	1 s
Packet queue check interval	0.5 s
Neighbor hold time	4 s
Transmission range	27800 m
Max packet queue length	10000 packets for ferry mode and 64 packets for buffer mode
Max packet queue hold time	1000 s for ferry mode and 30 s for buffer mode
Ferry sending rate	16 kb/s

The specifics of the AeroRP parameters listed in Table 3.7 are straightforward. In buffer mode, the packet queues are configured to be of similar size and timeout as AODV and DSDV. In ferry mode, the queue size and timeout are set to large values so that as many packets as possible can be ferried when AeroRP is in ferry mode. The rate at which the ferried or buffered packets are sent to the next hop or destination is configured to be twice that of the normal data sending rate so that these packets are distributed more quickly when a better node comes within transmission range.

3.4.2 Mobility Models

The simulations were tested with two different mobility models:

1. Random waypoint

2. Gauss-Markov

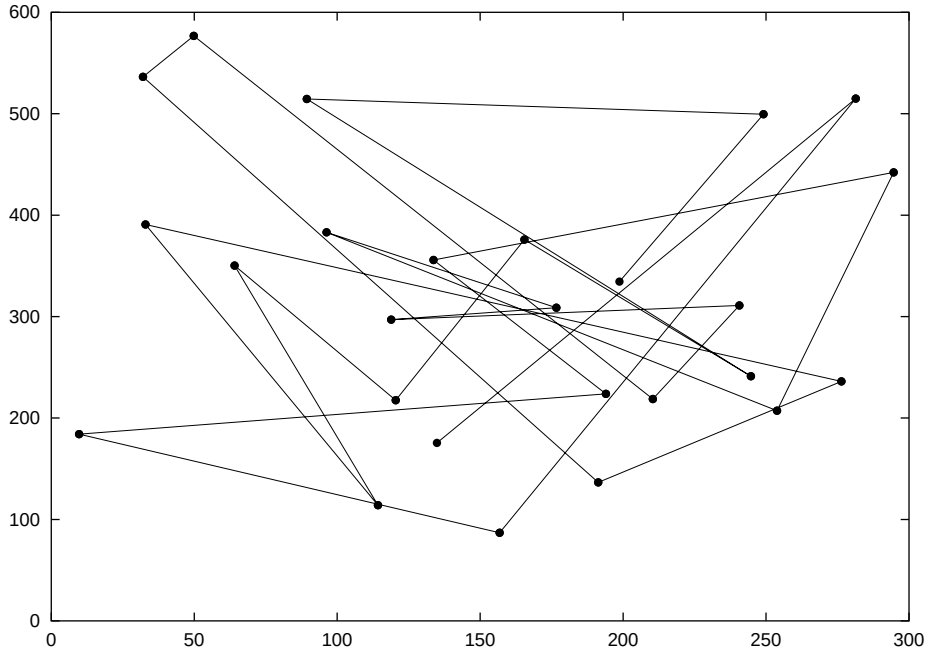

Figure 3.29. Random waypoint traveling pattern (reproduced from [6])

The random waypoint mobility model, which is already implemented in ns-3 in 2D, is discussed in [6]. The basic concept behind random waypoint is choosing a random destination, traveling to the destination, and then repeating the process. There is typically a pause time between direction and speed changes. However, since the AeroRP simulations are meant to mimic aircraft traveling patterns, the pause time is set to 0 as seen in Table 3.8. Figure 3.29 shows the traveling pattern for a single node moving per the random waypoint mobility model. A video [54] is available of 60 nodes moving using the random waypoint mobility model with 1200 m/s and 0 pause time. This video was generated directly from the ns-3 experiments conducted and used for visual verification of the mobility model.

The Gauss-Markov mobility model is also described in [6]. One of the main premises behind this mobility model is the ability to control the randomness (via

Table 3.8. Random waypoint parameters

Parameter	Value
Speed	Varies based on Table 3.2
Pause	0 s

α), and have the next change in direction and speed based on the previous direction and speed. Random waypoint results in the sharp changes in direction in Figure 3.29. However, Figure 3.30 shows a node adhering to the Gauss-Markov mobility model and the changes in direction are more smooth.

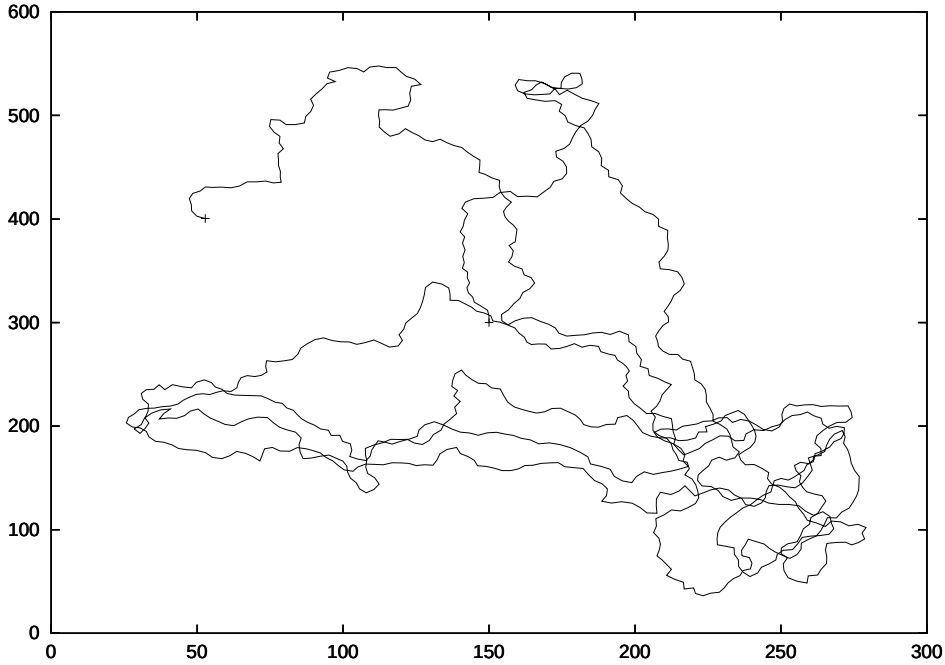

Figure 3.30. Gauss-Markov traveling pattern (reproduced from [6])

The Gauss-Markov mobility model is not currently part of the ns-3 distribution. Given that this mobility model more accurately mimics the mobility of aircraft due to the memory of its previous velocity, we used a high speed aircraft oriented (3D) Gauss-Markov mobility model for ns-3 created by the ResiliNets group at the University of Kansas [35]. The following explains the configurable

parameters for the Gauss-Markov mobility model implementation:

1. **Bounds** – 3D box that bounds the nodes
2. **TimeStep** – frequency in which direction, pitch, and velocity are changed
3. α – value between 0 and 1; the closer to 0, the more random the Gauss-Markov calculation will be whereas the closer to 1, the more memory from the previous calculation will be used so that it is not so random
4. **MeanVelocity** – initial mean velocity, which can be a constant or a range of velocities
5. **MeanDirection** – range in radians that indicates what direction a node will initially travel; for example, a range between 0 radians and 2π radians means the node can start heading off in any direction
6. **MeanPitch** – range in radians that indicates the up and down pitch of the node in the Z direction; for example, a range between –0.3 radians and 0.3 radians would translate to nose down 17 degrees to nose up 17 degrees
7. **NormalVelocity** – random Gaussian distribution that consists of three components that indicate the mean, standard deviation, and maximum deviation of the velocity
8. **NormalDirection** – random Gaussian distribution that consists of three components that indicate the mean, standard deviation, and maximum deviation of the direction in radians
9. **NormalPitch** – random Gaussian distribution that consists of three components that indicate the mean, standard deviation, and maximum deviation of the pitch in radians

Table 3.9. Gauss-Markov parameters

Parameter	Value
Time step	10 s
α	0.85
Mean velocity	Varies based on Table 3.2
Mean direction	Between 0 and 2π
Mean pitch	Between –0.03 and 0.03
Normal velocity	0 mean, 0 stdv, 0 max deviation
Normal direction	0 mean, 0.2 stdv, 0.4 max deviation
Normal pitch	0 mean, 0.01 stdv, 0.02 max deviation

Table 3.9 lists the Gauss-Markov mobility model values used in the simulation. The velocity values are chosen such that the velocity that the node starts with will be the same throughout the simulation, although the x, y, and z speed components of the velocity will change. The direction values are chosen such that a node can start moving in any possible direction of which the node can reasonably deviate from so that it is not always traveling in the same direction. Finally, the pitch values were chosen such that the node would only pitch up and down a maximum of 17 degrees of which it cannot deviate much from in order to reasonably limit mobility in the z direction.

A 2D video [55] is available of 60 nodes using the the Gauss-Markov mobility model, a constant 1200 m/s velocity, and parameters defined in Table 3.9. This video was generated directly from the ns-3 experiments conducted and used for visual verification of the mobility model.

Chapter 4

Analysis

This chapter contains the results and analysis of running the simulations detailed in Chapter 3. The analysis is broken down into effects of different mobility models in Section 4.1, different velocities in Section 4.2, different node densities in Section 4.3, and a detailed look at performance over time for AeroRP in the beginning of simulations in Section 4.4.

Various metrics were recorded for each simulation in order to produce the plots and analysis. The following are these metrics described in detail:

1. **Packet delivery ratio (PDR)** – The number of packets received divided by the number of packets sent at the application layer. Note that not necessarily all packets sent at the application layer will be sent at the MAC layer. This can happen if there is no route for the packet.

2. **Accuracy** – The number of packets received divided by the number of packets sent at the MAC layer. This allows us to measure how *accurate* a route is for a given routing protocol based on whether or not the route that was chosen for the packet results in a successful reception at the destination.

This is a good metric to gauge the quality of a route in a highly dynamic topology in which the validity of a route can rapidly change.

3. **Overhead** – The excess Bytes used to get the actual packet payload from source to destination. This includes the size of the control packets in the various routing algorithms as well as the size of the IP, UDP, and extra header information detailed in Section 3.3.2 for AeroRP in beaconless promiscuous mode detailed in Section 3.3.1.1.

4. **Delay** – The difference in time between when the originator of a data packet transmits the packet at the MAC layer and the time that the MAC layer of the final destination receives the data packet.

The plots in the following sections detail the above metrics and contain confidence interval bars at the points in the plot. Since each simulation is run 10 times, the 95% confidence intervals are calculated using a t-distribution [56]. This is calculated as $M \pm A \times \frac{s}{\sqrt{n}}$ in which M is the mean, A is the t-distribution value, s is the standard deviation, and n is the number of simulation runs for each point. The t-distribution value is 2.23 for 10 simulation runs with a 95% confidence. Note that some points may seem to not have any confidence interval bars. This is because they are too small to be seen on the plot. This indicates a higher confidence in the values that make up the mean.

4.1 Mobility Models

All simulations were run with both the random waypoint and Gauss-Markov mobility models discussed in Section 3.4.2. Analysis comparing the two mobility models showed that relative increase and decrease in PDR was similar for changes

in the simulation with variables such as node density. However, the Gauss-Markov mobility model always yielded lower PDR. This is illustrated in Figure 4.1 for different velocities and Figure 4.2 for different node densities. For example, observe the two AeroRP lines in Figure 4.2 to see that node density with both random waypoint and Gauss-Markov affects PDR similarly but that the PDR is lower with Gauss-Markov.

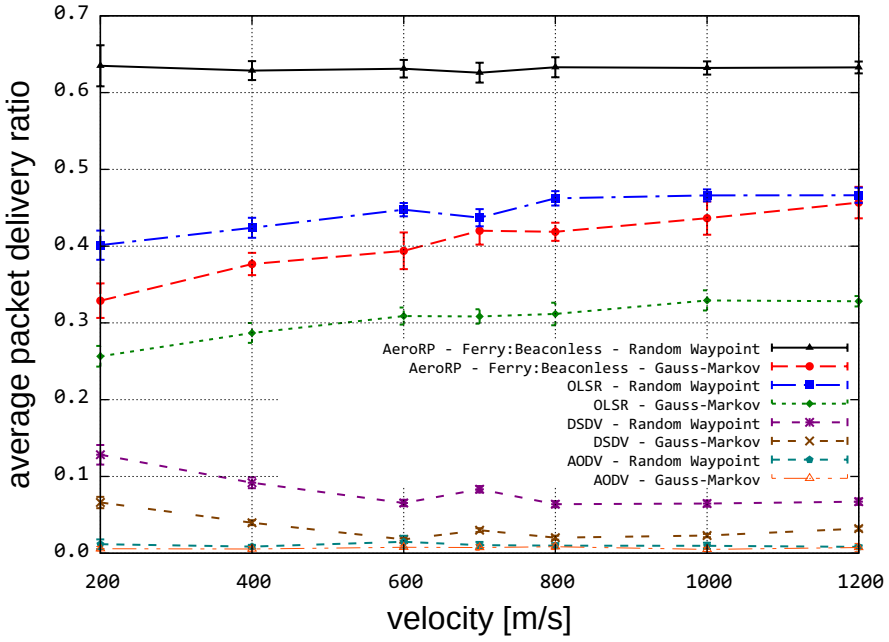

Figure 4.1. Effect of mobility model on PDR (60 nodes)

Basic performance testing was conducted to confirm that the ns-3 simulator correctly honors transmission ranges when separating nodes by a specific distance in 3D space as when nodes are separated by the same specific distance in 2D space. The nodes in random waypoint seem to make it all around the grid, but the the nodes in Gauss-Markov can be isolated to certain sections of the grid if they hit the boundary close to 90 degrees, causing nodes to bounce back close along their previous path. Sample movements of 4 nodes from a 60 node simulation

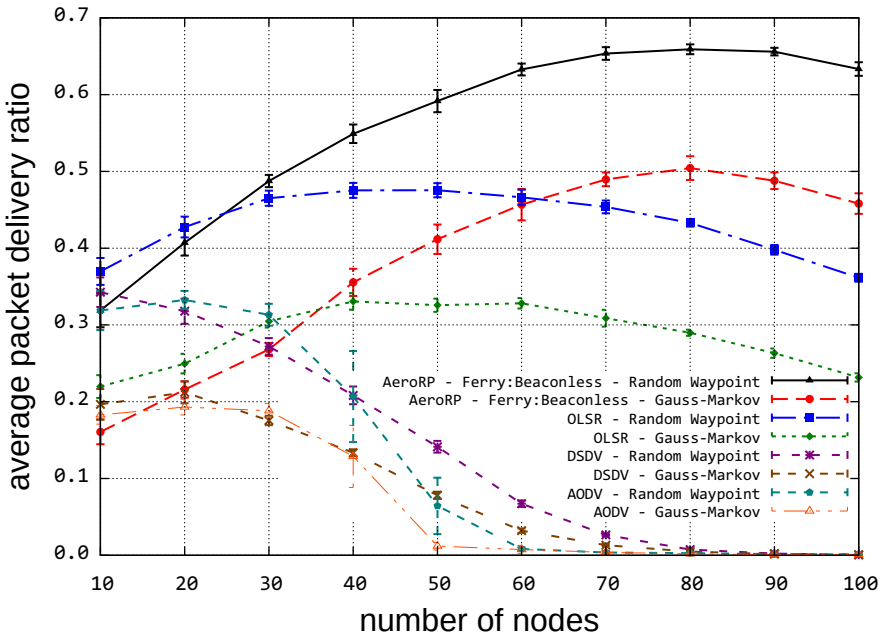

Figure 4.2. Effect of mobility model on PDR (1200 m/s)

for random waypoint is shown in Figure 4.3 and in Figure 4.4 for Gauss-Markov. Specifically, one can see in Figure 4.4 the blue and red lines that show 2 nodes being isolated to about half of the grid. Nodes in this pattern can introduce interference as well as reduce connectivity.

Another factor that is different between the two mobility models is that the random waypoint model operates only in 2D whereas the ResiliNets Gauss-Markov model operates in 3D space allowing the nodes to move up to 1000 m in the z-direction based on the parameters configured for the simulation bounds in Table 3.3. In random waypoint, nodes that are 27800 m apart in the xy-plane would be within transmission range of each other. However, if these same nodes were 27800 m apart in the xy-plane but differed by 1000 m in the z-direction with Gauss-Markov, they would not be within transmission range of each other. However, this is an extreme example.

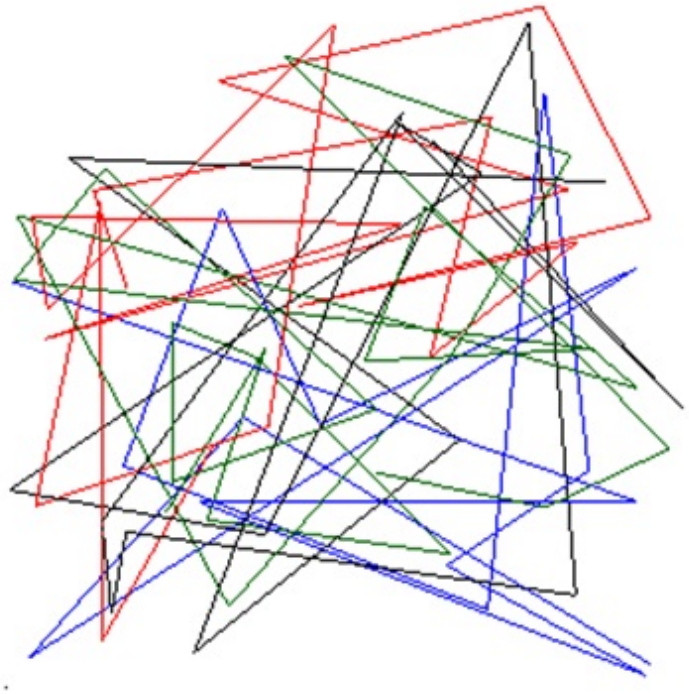

Figure 4.3. Traveling pattern of nodes with random waypoint

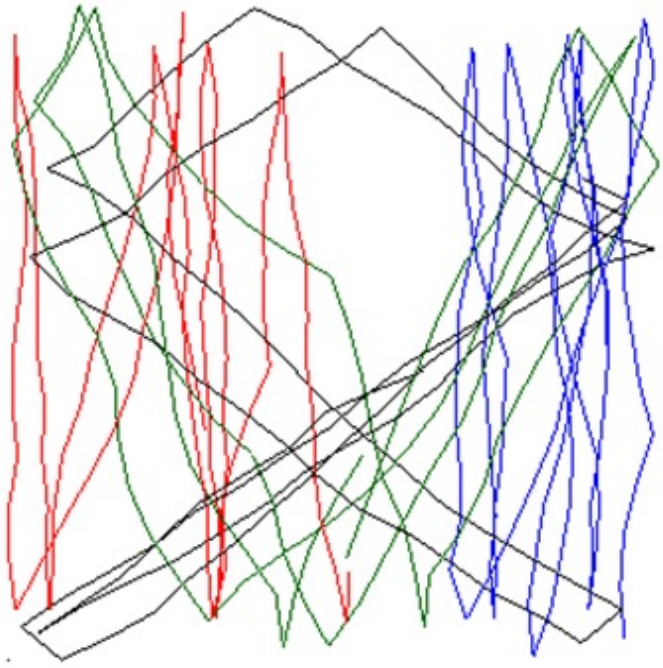

Figure 4.4. Traveling pattern of nodes with Gauss-Markov

Figure 4.5 shows the actual number of packets that were transmitted for different routing protocols. Packets are only transmitted when a route is found. Note that the routing protocols did not transmit as many packets in the Gauss-Markov mobility model because known routes were not quite as common as with the random waypoint mobility model.

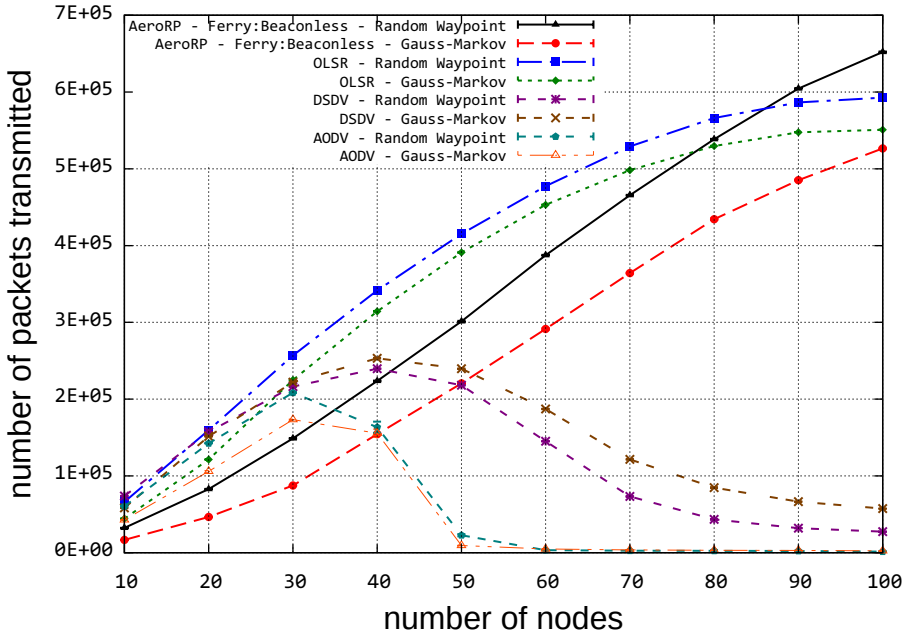

Figure 4.5. Effect of mobility model (1200 m/s)

The current implementation of the Gauss-Markov mobility model with aeronautical movements in mind is reasonable and more realistic, as far as aeronautical movement, when compared to the random waypoint mobility model. However, given that the random waypoint mobility model is a well studied and prevalent mobility model throughout networking literature, and the testing with random waypoint yielded better performance in this research, the rest of the analysis in this chapter will be with the random waypoint mobility model data. The complimentary plots for the following Section 4.2, Section 4.3, and Section 4.4, but with

the Gauss-Markov mobility model, are available in Appendix A.

4.2 Effects of Velocity

The effect of velocity in a MANET can be both beneficial and harmful; beneficial in the sense that the higher velocity gives nodes more of a chance to come in contact with each other, while at the same time being more harmful in the sense that routes become unstable and unreliable as the connectivity between nodes becomes shorter as the velocity increases.

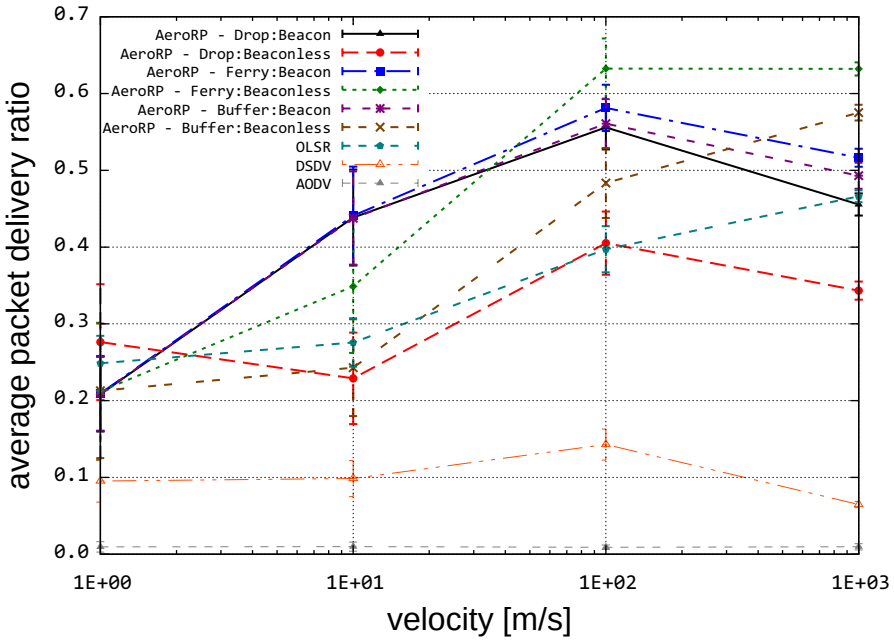

Figure 4.6. Effect of exponentially increasing velocity on PDR (60 nodes)

Figure 4.6 shows the performance in terms of PDR as the velocity exponentially increases from 1 m/s to 1000 m/s for 60 nodes. AODV did not perform well at higher node densities, including 60 nodes, as will be further discussed in Section 4.3. DSDV performs better than AODV but still poorly at this node density

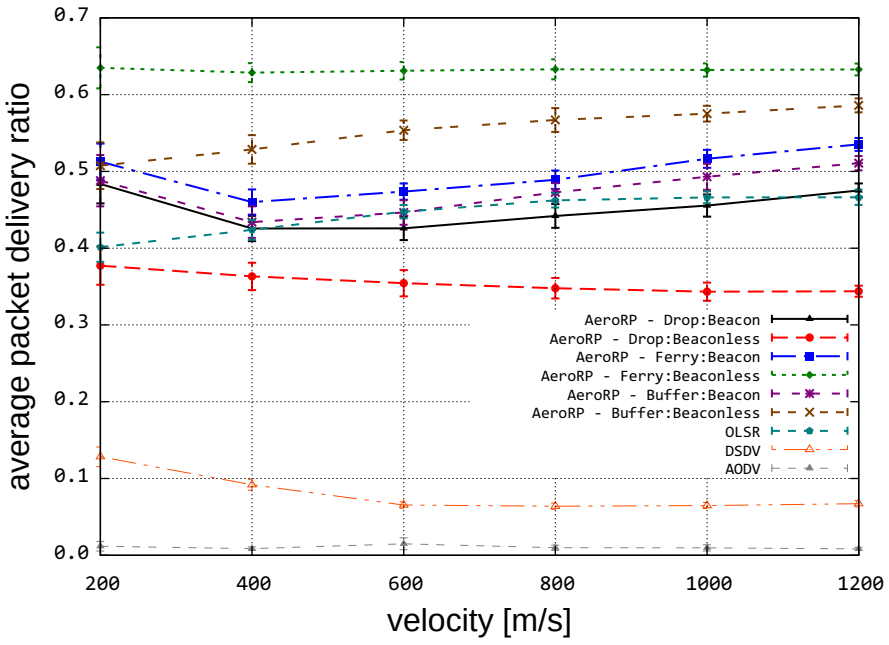

Figure 4.7. Effect of high velocity on PDR (60 nodes)

with its PDR approaching 0 as the velocity reaches 1000 m/s. OLSR performs well and in fact has higher PDR for all velocities when AeroRP is not ferrying and in beaconless promiscuous mode and for lower velocities when AeroRP is buffering packets in beaconless promiscuous mode. However, all other AeroRP modes outperform OLSR. Most of the routing protocols' PDR performance degrades as the velocity increases with the exception of OLSR and AeroRP when buffering packets in beaconless promiscuous mode. The plot clearly shows that ferrying packets in beaconless promiscuous mode outperforms everything and stays at a constant PDR when advancing from 100 m/s to 1000 m/s. The combination of some kind of packet buffering, whether it is indefinite or finite, coupled with beaconless promiscuous mode yields the best PDR.

Figure 4.7 is similar to Figure 4.6 but only shows higher velocities from 200 m/s to 1200 m/s and its effect on PDR. DSDV and AODV are again poor performers

in this scenario. All of the other routing protocols have fairly constant PDR values at these higher rates and most actually show an increase in performance as the velocity increases. This suggests that the higher velocities are actually improving the performance via connecting with more nodes.

Figure 4.8 shows how accuracy is affected as velocity increases exponentially from 1 m/s to 1000 m/s. Accuracy is a unique metric to measure in a highly dynamic topology. It gives one the notion of how accurate a route is at the time the packet is sent. Often, a route may be invalid in a highly dynamic scenario when the packet is sent and thus never reaches the destination. OLSR, DSDV, and AODV yield an accuracy of less than 50% except for OLSR at 1000 m/s. All of the various modes of AeroRP have an accuracy of 50% or higher at all velocities. This illustrates AeroRP's ability to accurately predict the delivery of a packet based on the known transmission range and the known and predicted distance between the source and next hop. Of the AeroRP modes, the beaconless promiscuous mode is more accurate than the beacon modes for two reasons. First, the surrounding nodes overhear data packets and thus trajectory data every single time a packet is transmitted. This results in sharing trajectory information more often than sending out periodic `hello` beacons. Second, putting the control data in the actual data packets makes the communication more symmetric than sending separate control packets. A control packet that is 44 Bytes may be transmitted successfully to a neighbor. However, this does not neccessarily mean that a 1000 Byte payload plus the control overhead can neccessarily be successfully transmitted to that same neighbor, especially if that neighbor is on the edge of the transmission range. So, in beaconless promiscuous mode, the neighbor list is built from data packets that it has definitely received from a given neighbor in the past and are thus more

accurate.

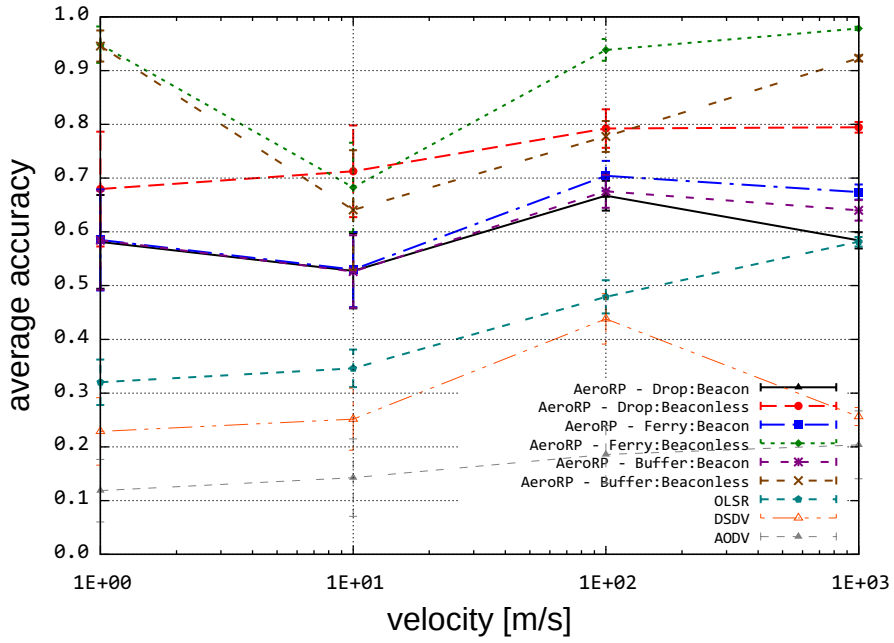

Figure 4.8. Effect of exponentially increasing velocity on accuracy (60 nodes)

Note that when looking at Figure 4.8, the accuracy for AeroRP when ferrying and buffering packets in beaconless mode significantly drops between 1 m/s and 10 m/s and then significantly increases between 10 m/s and 100 m/s. Figure 4.9 shows the same data but with more accuracy data points provided between 1 m/s and 10 m/s and between 10 m/s and 100 m/s. With this data, we can see that the accuracy immediately drops off between 1 m/s and 2 m/s, stays relatively constant between 2 m/s and 10 m/s, and then increases between 10 m/s and 100 m/s. The slow velocity at 1 m/s appears to have allowed AeroRP to have reasonably stable paths and lower overhead due to the beaconless mode which results in very good accuracy. However, doubling the velocity from 1 m/s to 2 m/s negatively effects most variations of AeroRP as the connectivity becomes more episodic. Nevertheless, the accuracy stays pretty constant after this drop

until the mobility is increased to 10 m/s at which point the accuracy for most AeroRP variations has increasing accuracy up to 100 m/s. Although AeroRP is designed with 100+ m/s velocities in mind, it still is more accurate than the other routing protocols at lower velocities.

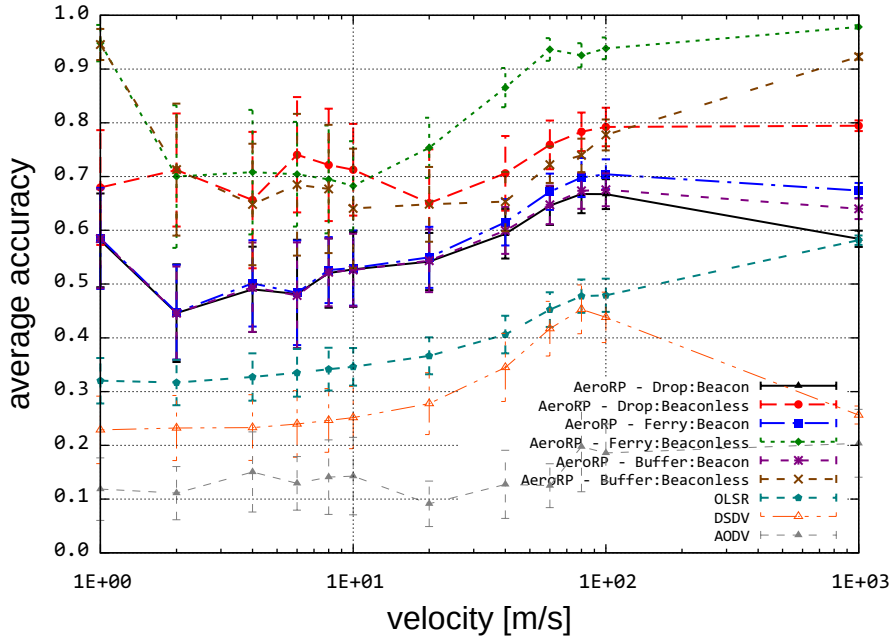

Figure 4.9. Effect of exponentially increasing velocity on accuracy (more detail – 60 nodes)

Figure 4.10 illustrates the same concept of accuracy as Figure 4.8 but only for the higher velocities from 200 m/s to 1200 m/s. Similar results are shown with all of the AeroRP modes outperforming OLSR, DSDV, and AODV at accuracies of 50% and higher. It's also important to note that AeroRP when ferrying packets in beaconless promiscuous mode has a steady accuracy of nearly 100%. This does not mean that the PDR is 100%, but it does show that the routes are almost 100% accurate when AeroRP does make the decision to route a packet.

The average overhead of the network measured in kb/s is shown in Figure 4.11 for exponentially increasing velocities from 1 m/s to 1000 m/s. OLSR and the

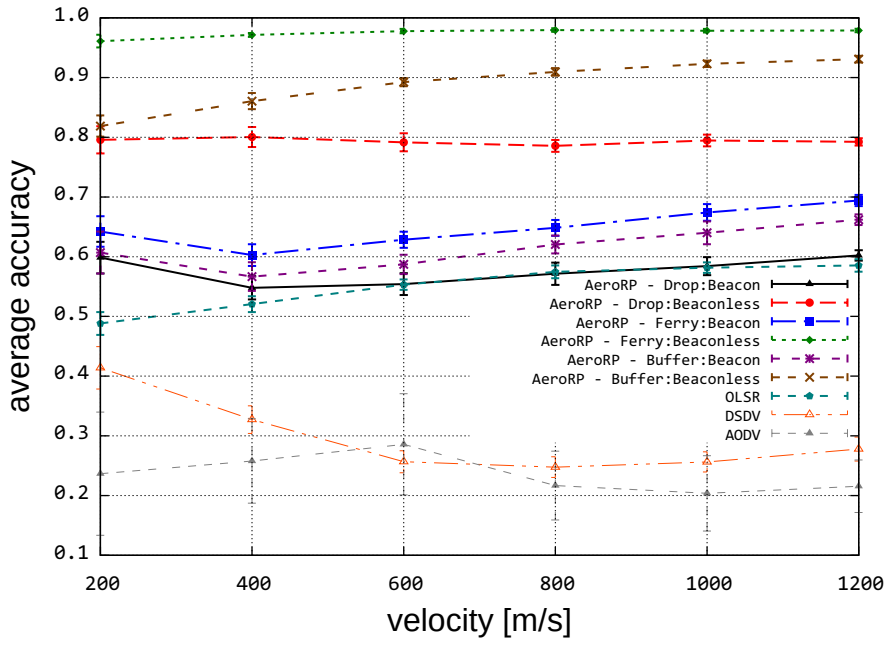

Figure 4.10. Effect of high velocity on accuracy (60 nodes)

different AeroRP modes have similar overhead footprints. The overhead of all the routing protocols stays relatively constant as the velocity increases starting with 100 m/s. This makes sense for OLSR and DSDV since they are proactive routing protocols and maintain their routes at regular intervals. This also makes sense for AeroRP since the overhead is constant and predictable in both beaconless and beacon modes. However, one would expect the overhead of AODV to change as the velocities increase causing more broken links and thus more overhead to find another route proactively. The AODV curve in the figure represents an approximate control overhead of 4.3 Mb/s, which is close to the maximum bandwidth that the network supports as discussed in Section 3.1. Saturating the network with control overhead at 60 nodes also explains why AODV yielded poor PDR as shown in Figure 4.6 and Figure 4.7. Figure 4.12 shows the same overhead behavior but only for the higher velocities from 200 m/s to 1200 m/s with the

overhead very constant at these higher velocities. Note that AeroRP in beaconless promiscuous mode with no ferrying has the lowest overhead. This makes sense because the beaconless promiscuous mode removes the need for separate control packets, thus cutting down on overhead. Also, this mode of AeroRP transmits less packets because it drops them immediately if there is no route, which also cuts down on overhead.

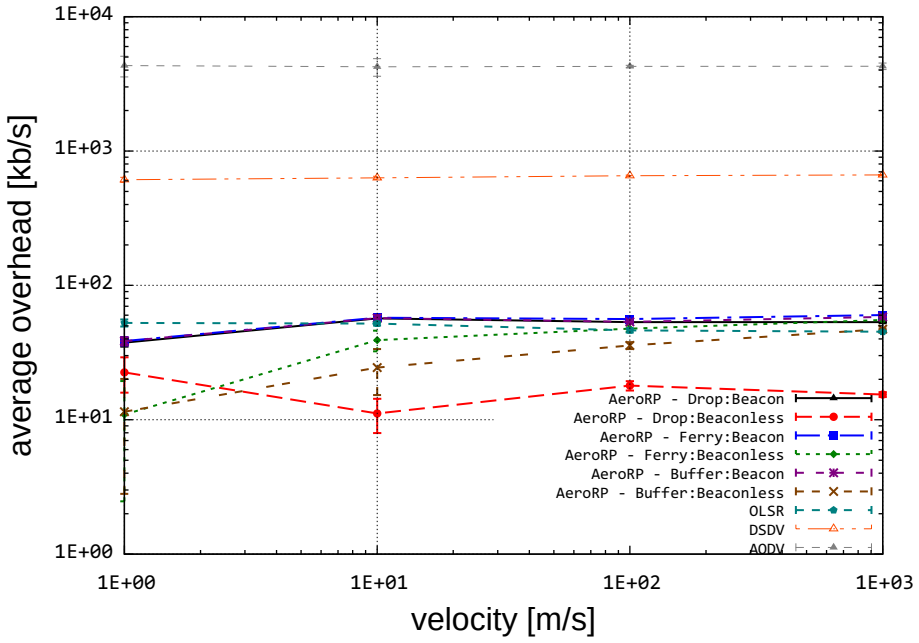

Figure 4.11. Effect of exponentially increasing velocity on overhead (60 nodes)

The average delay in ms between when a data packet is transmitted at the MAC layer of the originator of the packet and when the destination of the packet is received at the MAC layer is shown in Figure 4.13 for exponentially increasing velocities from 1 m/s to 1000 m/s. AeroRP when ferrying packets in beaconless promiscuous mode has the highest average delay at more than 10 s for the higher velocities. This makes sense since AeroRP, in this mode, will hold on to the packet as long as required and is generally able to successfully deliver more packets than

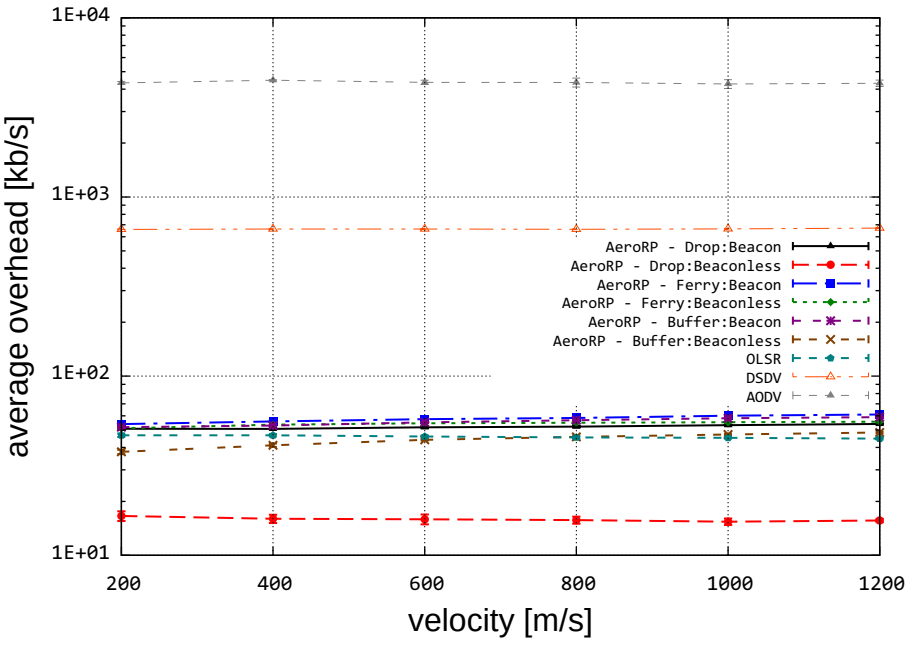

Figure 4.12. Effect of high velocity on overhead (60 nodes)

the other routing protocols as previously shown in Figure 4.6. The two AeroRP modes that buffer packets have delay in the middle compared to the other routing protocols because they will only hold packets for a finite amount of time. The two AeroRP modes that do not do any packet ferrying or buffering have the lowest delay because they will drop the packet immediately if there is no route. The trend for many of the routing protocols is for the delay to actually decrease as the velocity increases suggesting the nodes are transmitting packets more efficiently as they move more quickly. However, as Figure 4.14 illustrates, the delay stays relatively constant for all routing protocols at higher velocities of 200 m/s up to 1200 m/s.

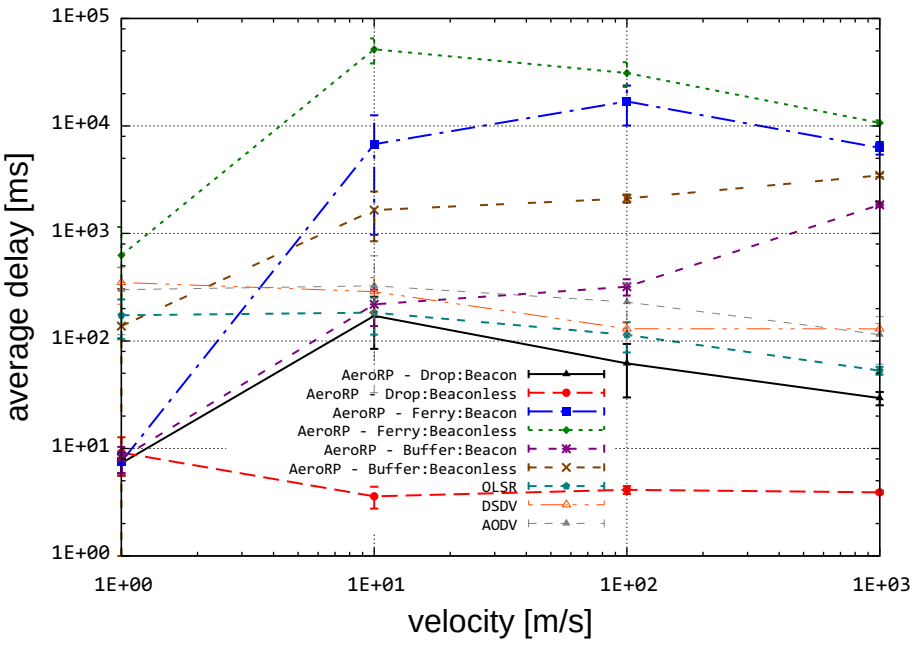

Figure 4.13. Effect of exponentially increasing velocity on delay (60 nodes)

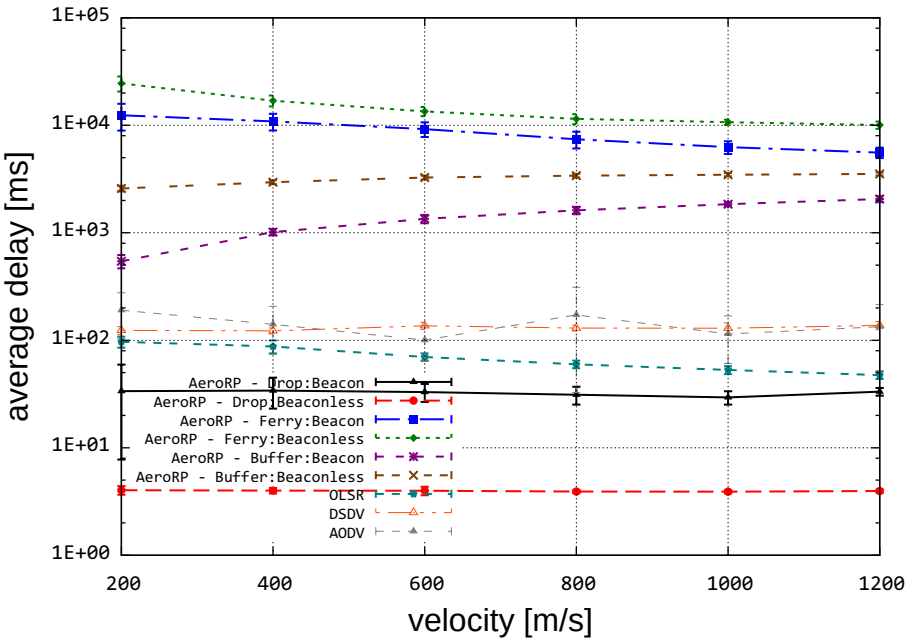

Figure 4.14. Effect of high velocity on delay (60 nodes)

4.3 Effects of Node Density

As with velocity, the increase in node density can be both beneficial and harmful; beneficial in the sense that more nodes means more hops and thus more connectivity; harmful in the sense that more nodes can cause interference and cause a network to degrade into the *parking lot problem* [47]. In this section, we look at the network performance when varying node densities from 10 to 100 nodes with constant high velocities of 1200 m/s as well as uniformly distributed velocities amongst the nodes between 200 m/s and 1200 m/s.

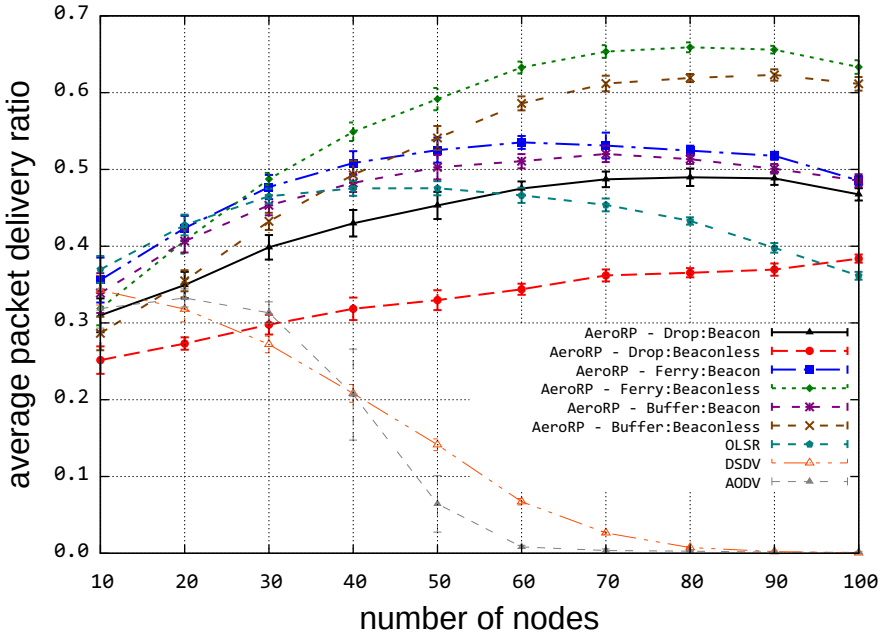

Figure 4.15. Effect of node density on PDR (1200 m/s)

Figure 4.15 shows the average PDR as the number of nodes are increased when traveling at 1200 m/s. The node density of the network affects all of the routing protocols with AeroRP ferrying packets in beaconless promiscuous mode performing the best. The PDR for all AeroRP modes increases as the number

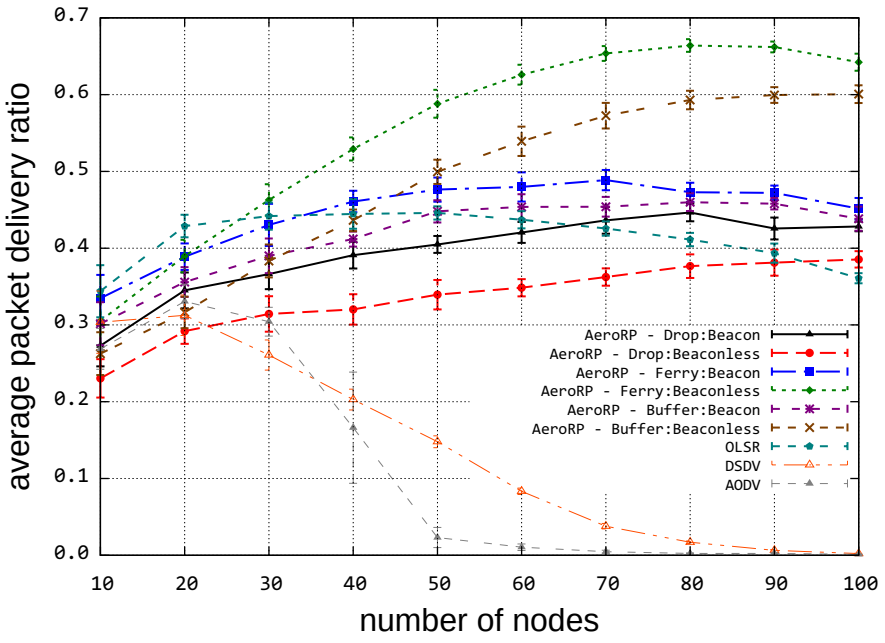

Figure 4.16. Effect of node density on PDR (200 ms/s - 1200 m/s)

of nodes increase with the exception of a slight performance degradation as the number of nodes approaches 90 and higher. The PDR for both DSDV and AODV immediately degrades as the number of nodes increases. This is most likely due to the increase in overhead as the number of nodes increases as described later in this section. The performance of OLSR starts to degrade around 50 nodes. This suggests, that as nodes increase, AeroRP is able to make more intelligent decisions on how to move the data packet towards the destination in the most optimal way whereas the non-AeroRP routing protocols are relying on non-geographic based links to move the packet to the destination. The same PDR data is shown in Figure 4.16 but the velocity uniformly distributed between 200 m/s and 1200 m/s. The lines are not as smooth but the overall results are basically the same.

Figure 4.17 shows how accurate the various routing protocols are in making it to the destination as the number of nodes increase when traveling at 1200 m/s.

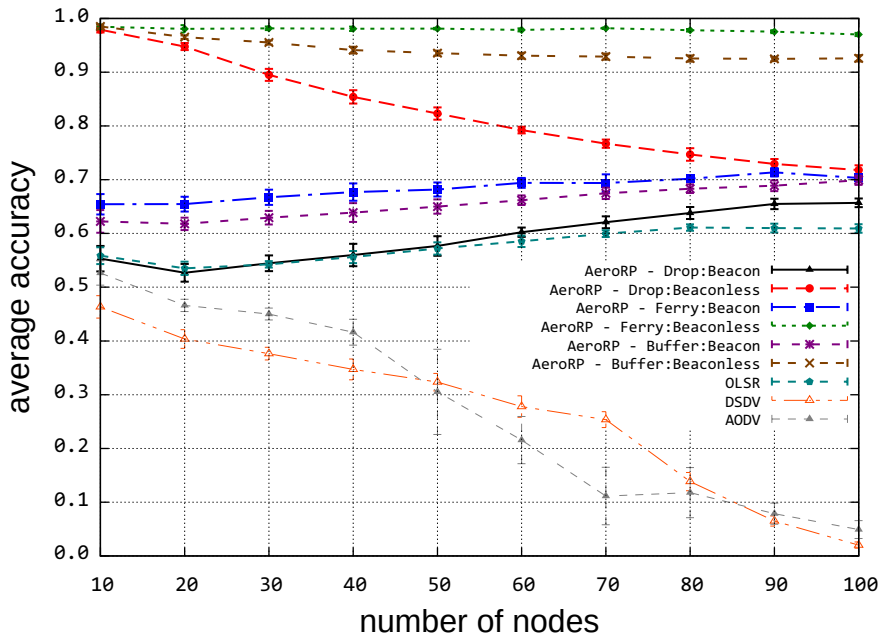

Figure 4.17. Effect of node density on accuracy (1200 m/s)

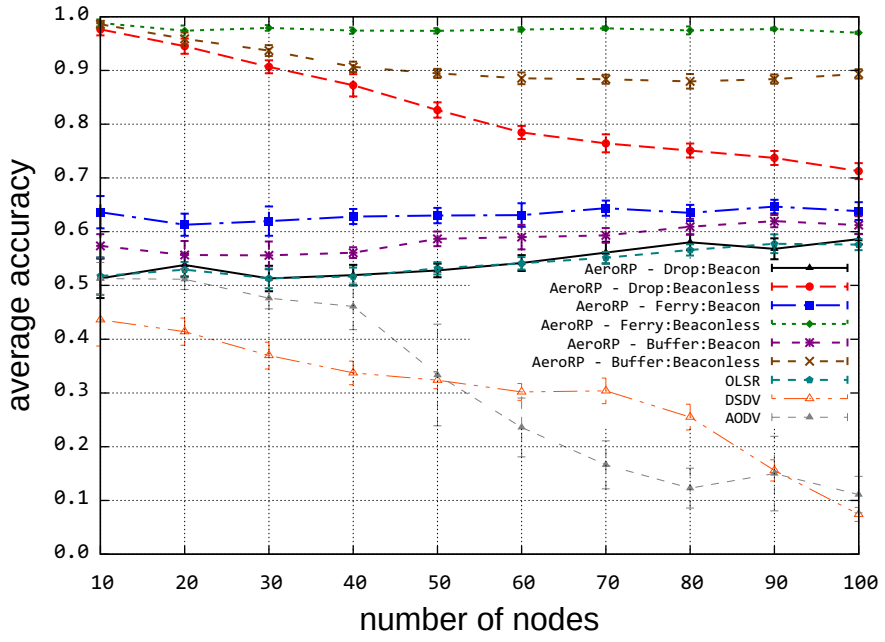

Figure 4.18. Effect of node density on accuracy (200 ms/s - 1200 m/s)

The accuracy of ferrying and buffering packets with AeroRP stays constant at almost 100% as the number of nodes increases. This high accuracy is due to the same reasons discussed in Section 4.2 when the velocities are increasing. All of the AeroRP modes have an accuracy of 50% or higher with the accuracy increasing as the number of nodes increase with the exception of AeroRP running in beaconless promiscuous mode but with no ferrying or buffering of packets. This decrease in accuracy can be attributed to the nodes having to rely on data transmissions to communicate their trajectories to nearby nodes. The buffering and ferrying allows data to be delivered at different times in the simulation in which AeroRP that is not ferrying or buffering packets is not sharing this information as often. This does not occur when AeroRP is in beacon mode but not ferrying or buffering packets because it is still regularly sharing its trajectory information with its neighbors in the form of periodic `hello` beacons. OLSR yields higher accuracy as the number of nodes increases but still not as high as the AeroRP modes. The accuracy of DSDV and AODV actually decrease as the number of nodes increase, probably due to the increase in overhead as the number of nodes increases as described next in this section. Figure 4.18 shows this same data but with the velocity of the nodes distributed between 200 m/s and 1200 m/s. Again, the performance is basically the same but the accuracy lines are not quite as smooth when compared to the constant 1200 m/s velocity.

As expected, and illustrated in Figure 4.19, the average overhead of the network increases with the node density. However, the proactive discovery and maintenance of routes in AODV results in exponentially increasing overhead as the number of nodes increase from 30 to 50 nodes. The overhead of AODV would probably continue to exponentially increase and monopolize the bandwidth if it

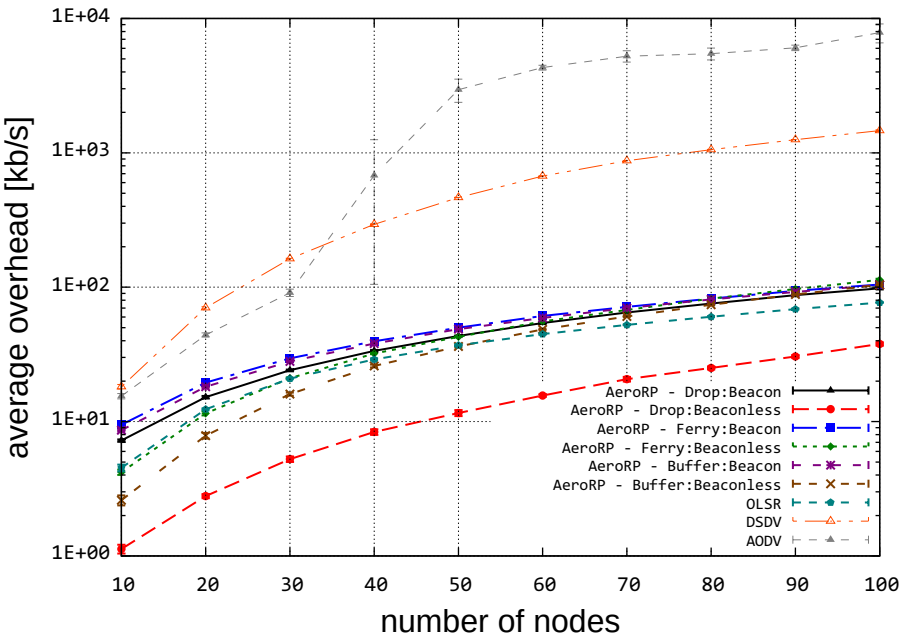

Figure 4.19. Effect of node density on overhead (1200 m/s)

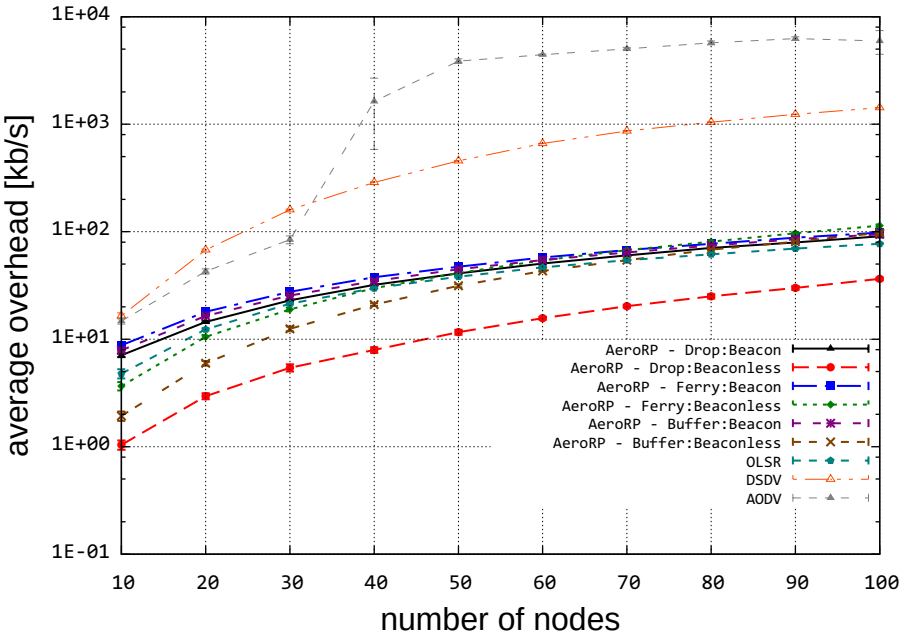

Figure 4.20. Effect of node density on overhead (200 ms/s - 1200 m/s)

did not hit the network saturation point. The overhead of DSDV increases linearly from 18 kb/s at 10 nodes to 1.5 Mb/s at 100 nodes. The overhead of OLSR and the AeroRP routing protocols also increase linearly, but not as drastically as DSDV, from around 15 kb/s at 10 nodes to around 100 kb/s at 100 nodes. Note that AeroRP in beaconless promiscuous mode with no ferrying has the lowest overhead for the same reasons described in Section 4.2. With overhead, we again see that the behavior of distributing the velocities between 200 m/s and 1200 m/s is similar to the nodes having a constant velocity of 1200 m/s as illustrated in Figure 4.20.

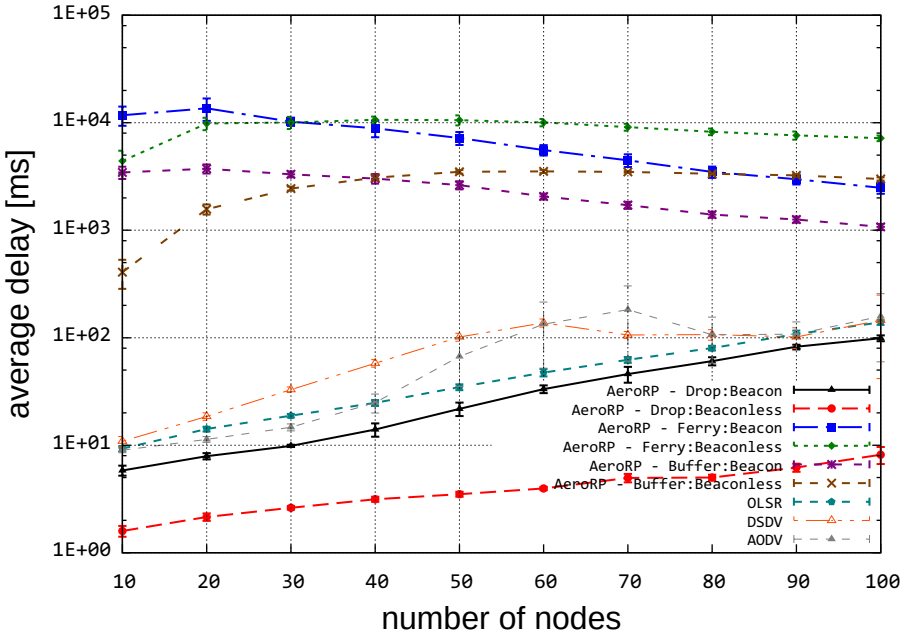

Figure 4.21. Effect of node density on delay (1200 m/s)

The effect that node density has on the delay of data packet transmissions for nodes moving at a constant velocity of 1200 m/s is shown in Figure 4.21. The AeroRP modes that ferry or buffer packets have the highest delay for the same reasons explained in Section 4.2. The AeroRP modes that immediately drop

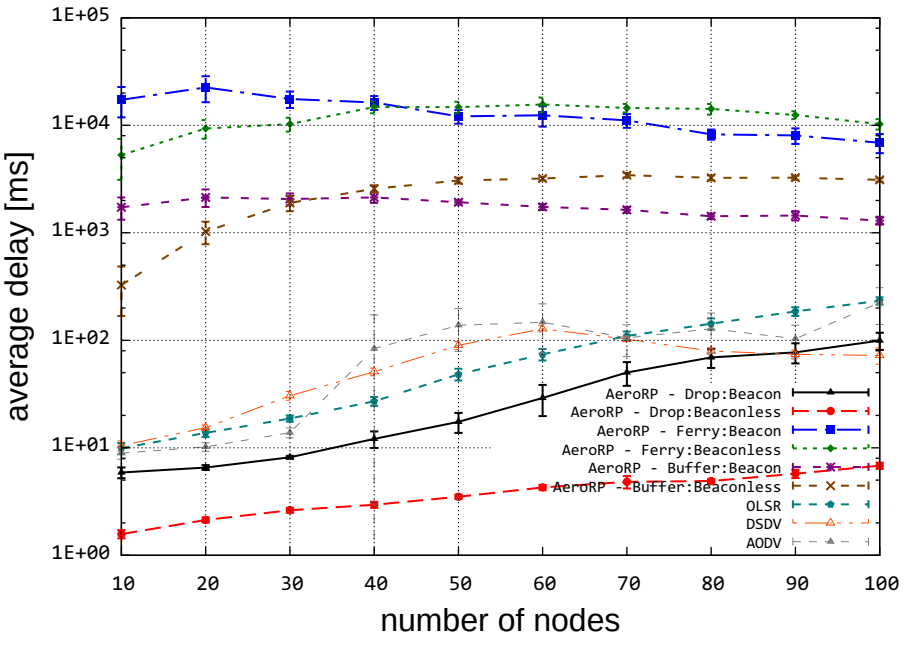

Figure 4.22. Effect of node density on delay (200 ms/s - 1200 m/s)

packets that do not have routes have the lowest delay of all of the routing protocols for the same reasons that are also explained in Section 4.2. It is interesting to note, however, that the ferrying and buffering modes of AeroRP actually decrease in delay as the node density increases while the other AeroRP modes, AODV, DSDV, and OLSR actually increase in delay as the node density increases. Perhaps as the node density increases, the buffering and ferrying modes are able to get their buffered and ferried packets to their destinations more quickly while the increase in nodes results in more hops and thus longer delay to the destination. It is suspected that calculating just the average delay for data packets that were not ferried or buffered for the AeroRP modes that do this would result in the same increasing trend of delay as the number of nodes increases. Finally, Figure 4.22 shows this same phenomenon but with slightly less smooth delay lines and the nodes varying their velocities between 200 m/s and 1200 m/s.

It should be noted that both the constant 1200 m/s velocity and varying velocities between 200 m/s and 1200 m/s plots were shown for the different performance metrics in this section to illustrate the effect of node density was similar for all higher velocities.

4.4 AeroRP Startup Analysis

In the previous sections of this analysis chapter, we looked at the performance metrics averaged across the entire duration of the simulation. In this section, we look at the PDR performance at the beginning of the simulation for each second of the simulation to see how the performance of the network behaves when initialized. Note that every simulation has a warmup time of 100 s before the actual data packet transmission begins.

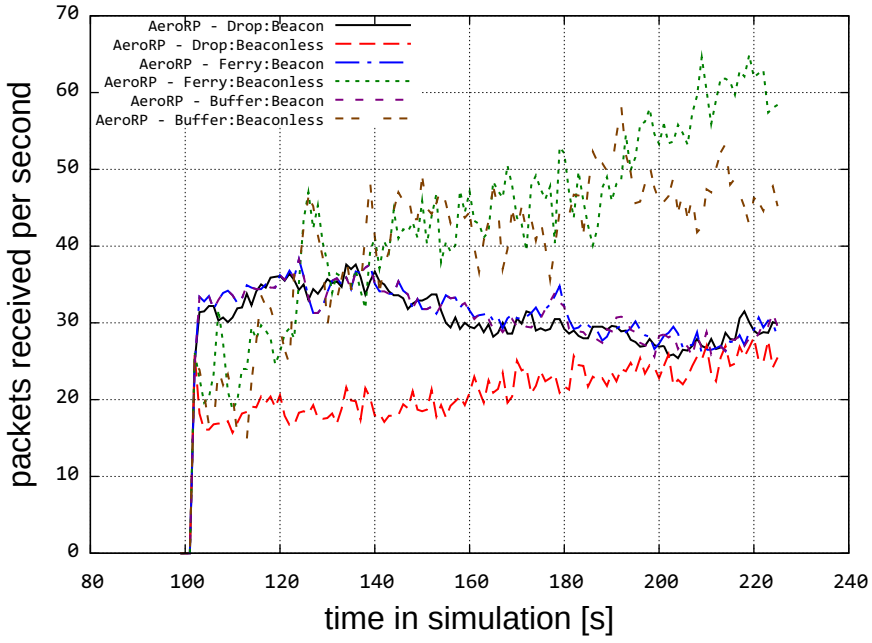

Figure 4.23. Detailed look at startup (200 m/s - 60 nodes)

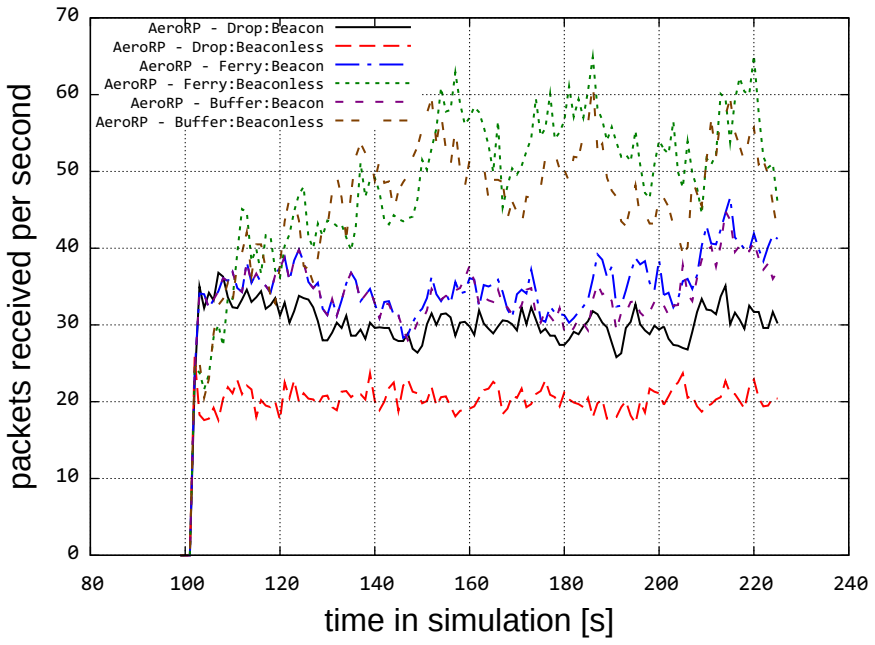

Figure 4.24. Detailed look at startup (1200 m/s - 60 nodes)

Figure 4.23 shows the number of packets being received per second for different AeroRP modes at the beginning of a simulation of 60 nodes that are traveling at 200 m/s. The AeroRP modes that buffer or ferry packets in beaconless promiscuous mode first start out at having around 20 packets received per second but eventually increase and stabilize around 50 and 60 packets received per second by 225 seconds into the simulation. The performance of these AeroRP modes gradually increase as they distribute more and more buffered and ferried packets amongst the nodes thus distributing more trajectory information amongst the nodes to make better routing decisions. The different AeroRP modes that use beacons to communicate trajectory information perform similarly amongst themselves. The AeroRP mode that does not ferry or buffer packets in beaconless promiscuous mode has the most trouble delivering packets and stabilizing because it is relying on overhearing nodes that have found routes and are transmitting data packets

in order to populate its neighbor to make routing decisions. This transmission of data is not as common since the nodes are not buffering or ferrying packets until a suitable neighbor or the destination is found. Figure 4.24 shows this same detailed look at the beginning of the simulation but with the nodes traveling at 1200 m/s as opposed to 200 m/s in Figure 4.23. The behavior is the same but the increase of packet reception and stabilization for the buffering and ferrying of packets in beaconless promiscuous mode occurs much quicker at around 150 seconds as opposed to 200 seconds due to the faster velocity.

Chapter 5

Conclusions and Future Work

This chapter summarizes this thesis, with conclusions in Section 5.1 and suggestions for future work presented in Section 5.2.

5.1 Conclusions

In this thesis, the geographic-based routing protocol for highly dynamic MANETs, AeroRP and its various modes, has been presented in detail. Significant effort has also been spent investigating the behavior of ns-3 and tuning and changing it to work with longer transmission ranges for the aeronautical use cases.

The contributions of this thesis are:

- Basic validation and performance testing of the new ns-3 network simulator [13]
- Model the AeroRP geographic based routing protocol
  - finalize and confirm the calculations used to make routing decisions
  - detail various modes, including ferrying and beaconless modes

- Implement AeroRP in the ns-3 network simulator
- Analyze the performance of AeroRP against other MANET routing protocols using the ns-3 network simulator

From the analysis chapter of this thesis, many things have been learned regarding the performance of AeroRP, its various modes, and other MANET routing protocols from the results of the simulations. The different routing protocols have tradeoffs for different velocities and node densities with respect to different performance metrics explored. The results of these simulations and their effects on the different performance metrics have been summarized in Figure 5.1. Note that the table is not a comment on how good or bad a specific metric is, but a relative comparison of the metric amongst the routing protocols simulated. Depending on the node density and velocity, this table can be used as a guide to choose a routing protocol depending on which metrics are most important for a given scenario.

AODV and DSDV do not perform very well in highly dynamic scenarios as far as PDR. OLSR was observed to perform well and in fact outperformed some of AeroRP's modes. For AeroRP, the combination of some kind of packet buffering, whether indefinite or finite, coupled with beaconless promiscuous mode yields the best PDR. Besides AODV and DSDV, the other routing protocols have fairly constant PDR values at higher velocities and most actually show an increase in performance as the velocity increases. This suggests that the higher velocities are actually improving the performance via connecting with more nodes. AeroRP is able to make more intelligent decisions on how to move the data packet towards the destination in the most optimal way whereas the non-AeroRP routing protocols are relying on non-geographic based links to move the packet to the destination.

OLSR and DSDV maintain the nodes' routing tables by sending the routing

			AODV	DSDV	OLSR	AeroRP					
						Beacon			Beaconless		
						Drop	Buffer	Ferry	Drop	Buffer	Ferry
Low Node Density	Low Velocity	PDR	Fair	Fair	Fair	Fair	Fair	Poor	Poor	Poor	Fair
		Accuracy	Fair	Fair	Fair	Fair	Fair	Fair	Good	Good	Good
		Overhead	Poor	Poor	Fair	Fair	Fair	Fair	Good	Good	Good
		Delay	Fair	Fair	Fair	Fair	Fair	Poor	Good	Good	Poor
	High Velocity	PDR	Fair	Fair	Good	Fair	Good	Good	Poor	Fair	Good
		Accuracy	Poor	Poor	Poor	Poor	Fair	Fair	Good	Good	Good
		Overhead	Poor	Poor	Fair	Fair	Fair	Fair	Good	Fair	Fair
		Delay	Good	Good	Good	Good	Fair	Poor	Good	Fair	Poor
HIgh Node Density	Low Velocity	PDR	Poor	Poor	Fair	Good	Good	Good	Fair	Good	Good
		Accuracy	Poor	Poor	Fair	Fair	Fair	Fair	Fair	Fair	Fair
		Overhead	Poor	Poor	Fair	Fair	Fair	Fair	Good	Fair	Fair
		Delay	Fair	Fair	Fair	Fair	Fair	Poor	Good	Poor	Poor
	High Velocity	PDR	Poor	Poor	Fair	Fair	Fair	Fair	Fair	Good	Good
		Accuracy	Poor	Poor	Fair	Fair	Fair	Fair	Fair	Good	Good
		Overhead	Poor	Poor	Fair	Fair	Fair	Fair	Good	Fair	Fair
		Delay	Fair	Fair	Fair	Fair	Fair	Poor	Good	Poor	Poor

Figure 5.1. Summary table of routing performance

data amongst the network in a proactive nature and do not necessarily expect an acknowledgment. However, AODV must make explicit route requests, RREQs, when it needs a route for which it does not currently have an active one for a given destination. It must also then get an explicit route reply back, RREP, in order to send data to that destination. The RREP may be sent back on a broken path due to the highly dynamic nature of the topology we are studying. A new RREQ will than have to be sent until the originator successfully receives a valid route. This can result in the poor performance and large control overhead that was observed from AODV in the simulations.

OLSR, DSDV, and AODV yield an accuracy much lower than all of the AeroRP modes. This illustrates AeroRP's ability to accurately predict the delivery of a packet based on the known transmission range and the known and predicted distance between the source and next hop. Better accuracy uses the network resources more efficiently.

For the beaconless AeroRP modes, putting the control data in the actual data packets makes the communication more symmetric than sending separate control packets. A control packet might be able to make it to a neighbor but that does not mean that a larger data packet can be successfully transmitted to that same neighbor. The neighbor list is built from data packets that it has definitely received from a given neighbor in the past and are thus more accurate.

AeroRP when ferrying packets in beaconless promiscuous mode has the highest average delay. In this mode, AeroRP will hold on to the packet as long as required and is generally able to successfully deliver more packets than the other routing protocols. The AeroRP buffering modes are a nice compromise between the higher packet delivery but higher delay of the ferrying modes and the lower

packet delivery but lower delay of the non-ferrying modes. The two AeroRP modes that do not do any packet ferrying or buffering have the lowest delay because they will drop the packet immediately if there is no route.

5.2 Future Work

As far as future work, there are several areas that can be investigated further, implemented, and improved upon:

1. The AODV implementation in ns-3 performs very poorly with respect to PDR. As shown in Chapter 4, this is most likely due to the exponentially increasing overhead that saturates the network. We are continuing to work with the ns-3 developers to improve the performance.

2. Integrate with AeroNP and AeroTP [57] when they have been implemented in ns-3. AeroNP is the aeronautical network protocol and AeroTP is the aeronautical transport protocol being developed by ResiliNets.

3. Work on additional modes of AeroRP. For instance, a mode in which the ground station periodically updates all of the airborne nodes with the trajectory information of all other nodes in the network that are predicted by the mission plan.

4. Investigate other recovery mechanisms for AeroRP when a route is not available besides ferrying, buffering, or dropping the packet.

5. Investigate security issues. For instance, how AeroRP can operate in situations in which trajectory information cannot be shared amongst the nodes due to security concerns.

6. Testing with AeroRP both with beaconless promiscuous mode and sending periodic `hello` beacons. Are there any advantages to this?

7. ns-3 is a relatively young network simulator compared to its mature predecessor, ns-2. As new routing protocols are developed for ns-3, there will be more to compare AeroRP to. The performance and tuning of ns-3 for optimal behavior of MANETs is an ongoing exercise.

Appendices

Appendix A

Gauss-Markov Plots

Section 4.2, Section 4.3, and Section 4.4 contain plots with data from simulations using the random waypoint mobility model. The following plots are equivalent but with using the data from simulations using the Gauss-Markov mobility model.

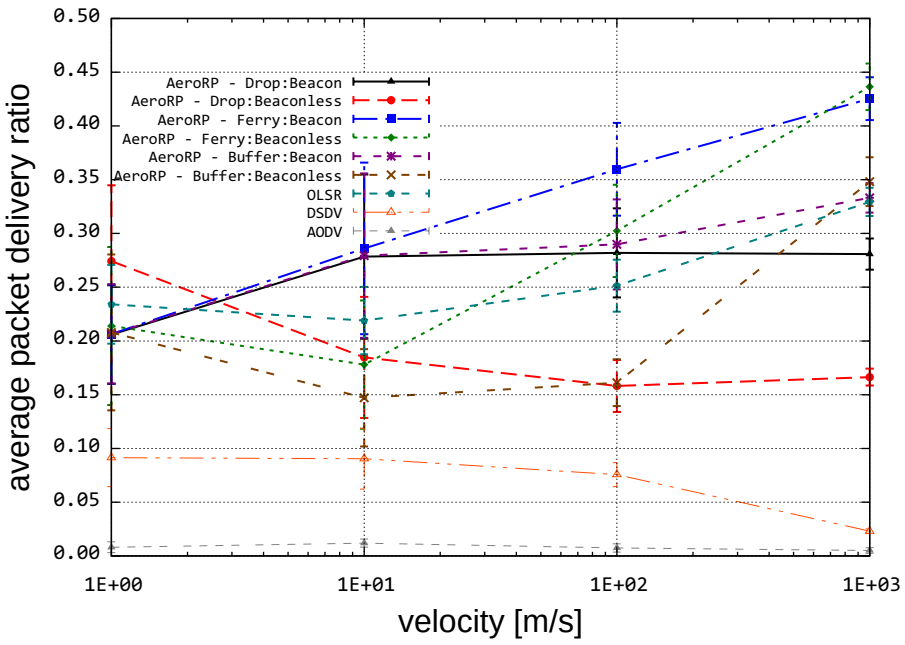

Figure A.1. Effect of exponentially increasing velocity on PDR (60 nodes)

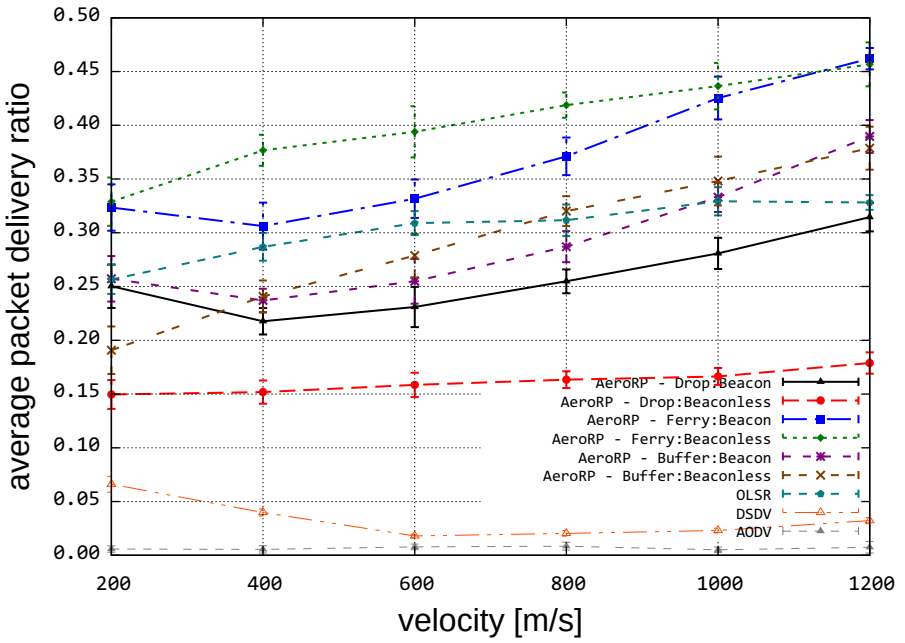

Figure A.2. Effect of high velocity on PDR (60 nodes)

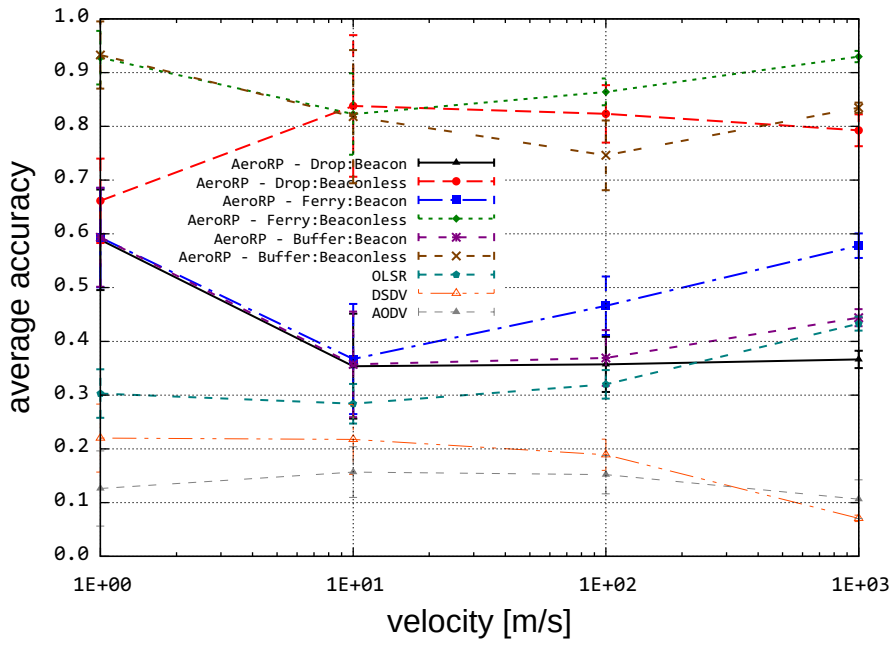

Figure A.3. Effect of exponentially increasing velocity on accuracy (60 nodes)

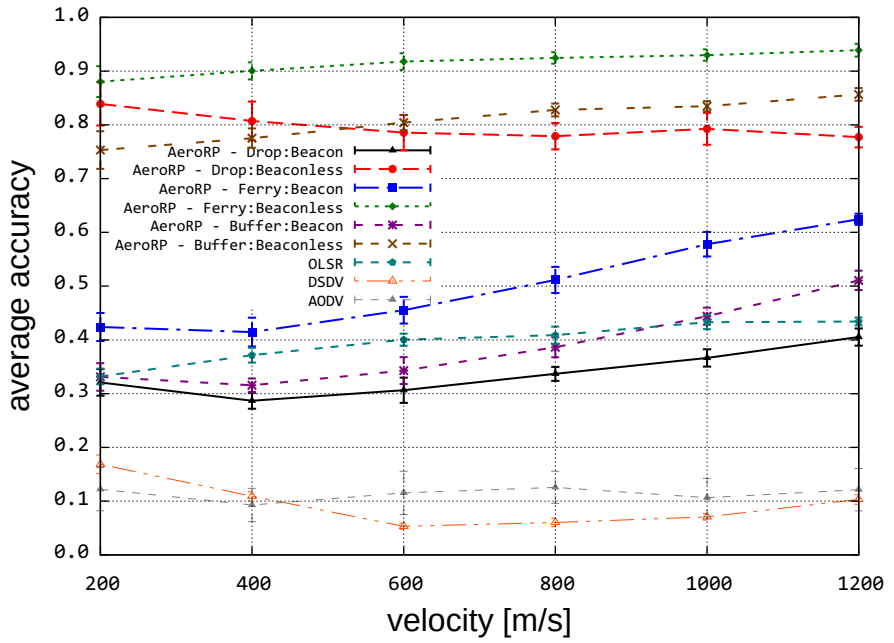

Figure A.4. Effect of high velocity on accuracy (60 nodes)

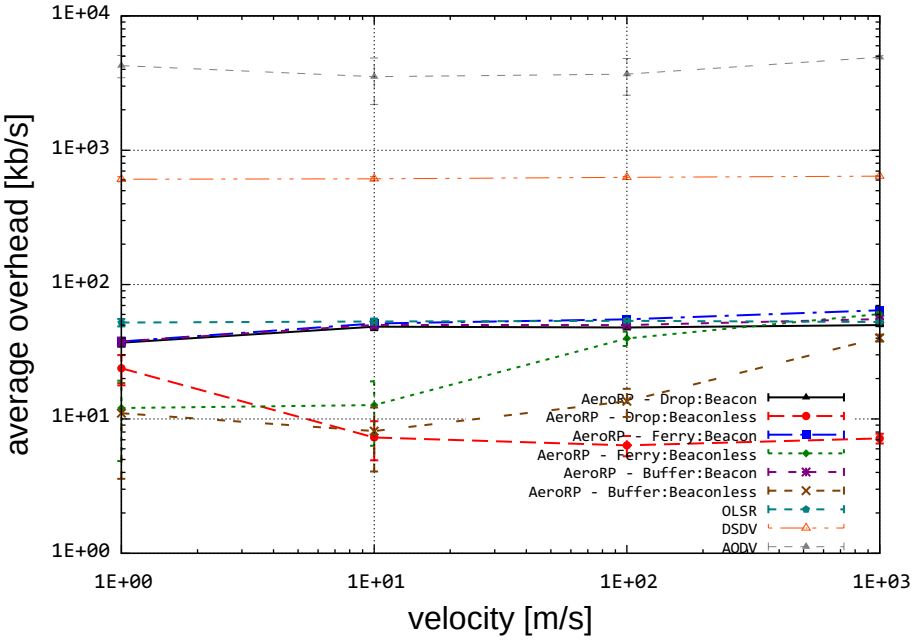

Figure A.5. Effect of exponentially increasing velocity on overhead (60 nodes)

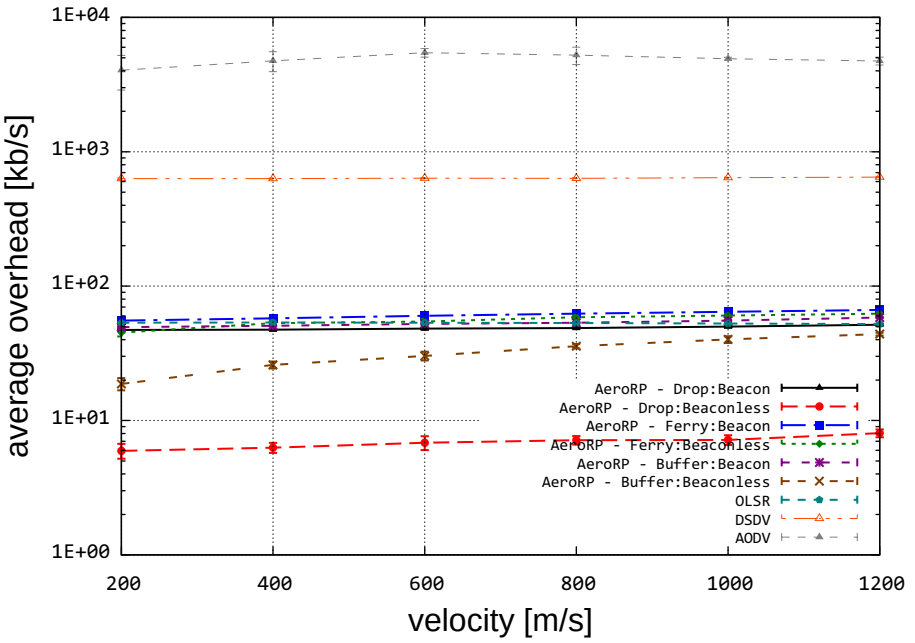

Figure A.6. Effect of high velocity on overhead (60 nodes)

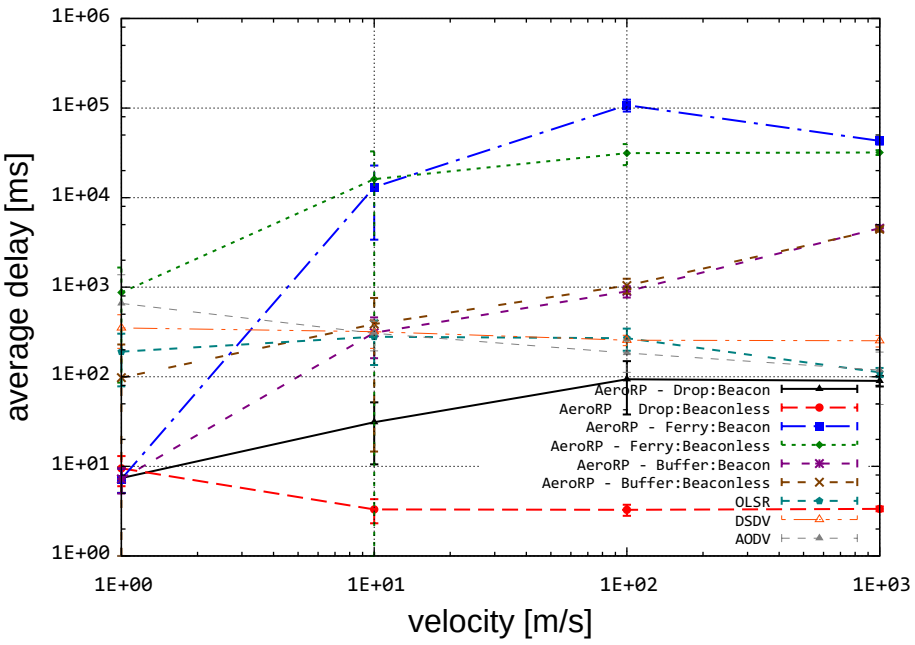

Figure A.7. Effect of exponentially increasing velocity on delay (60 nodes)

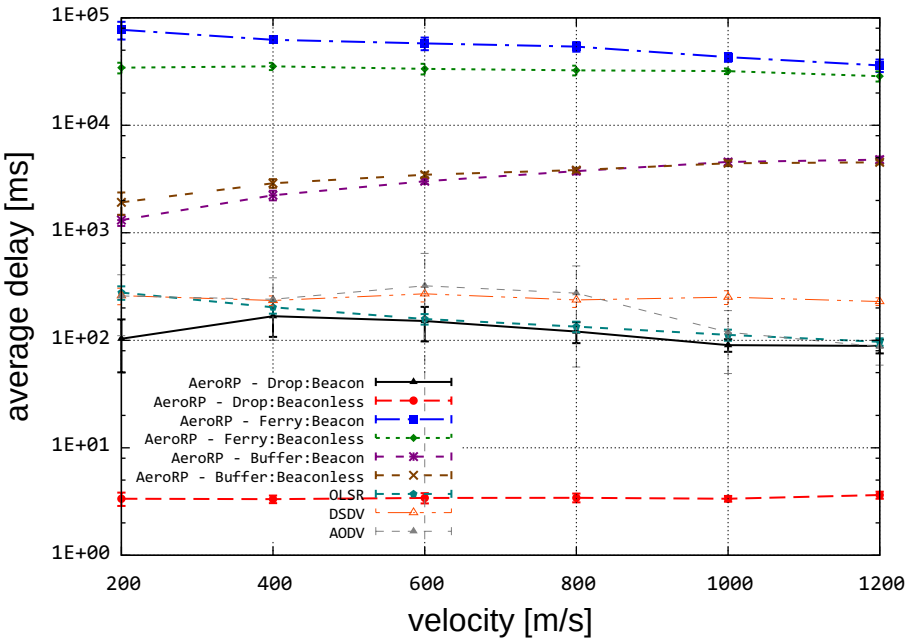

Figure A.8. Effect of high velocity on delay (60 nodes)

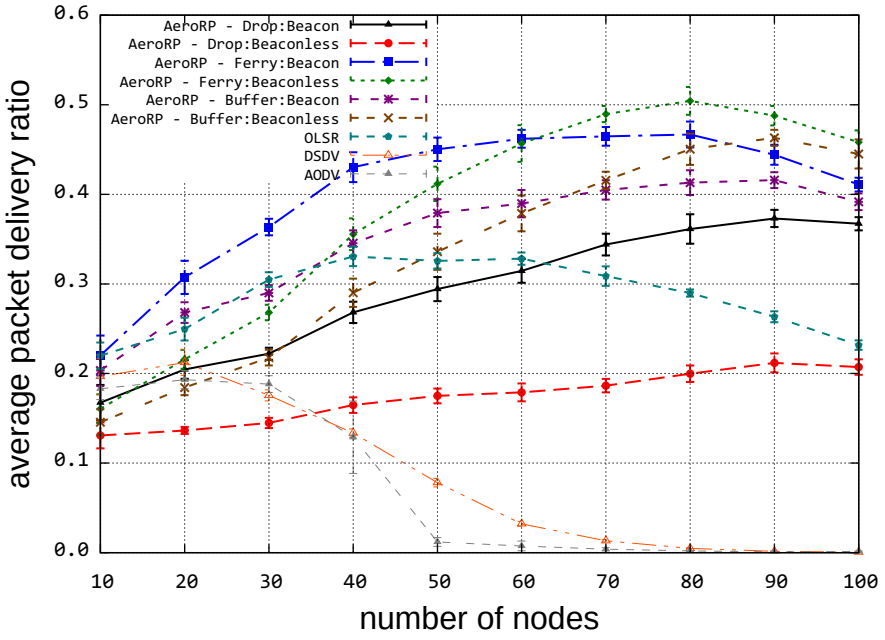

Figure A.9. Effect of node density on PDR (1200 m/s)

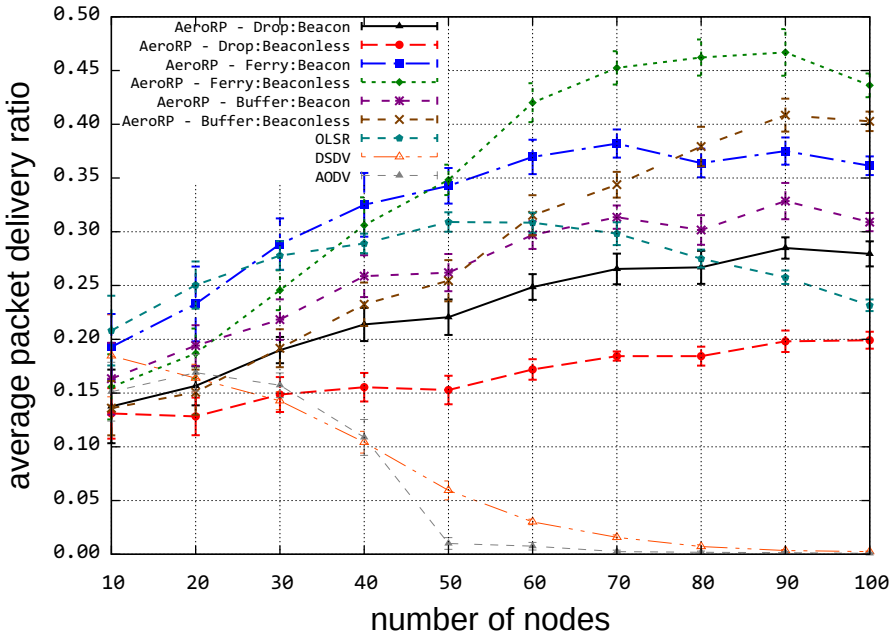

Figure A.10. Effect of node density on PDR (200 ms/s - 1200 m/s)

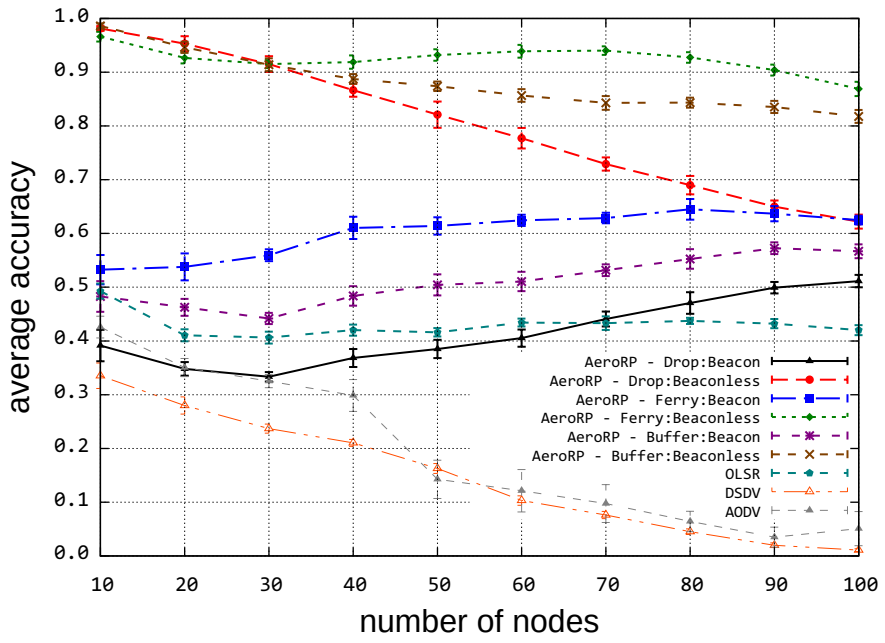

Figure A.11. Effect of node density on accuracy (1200 m/s)

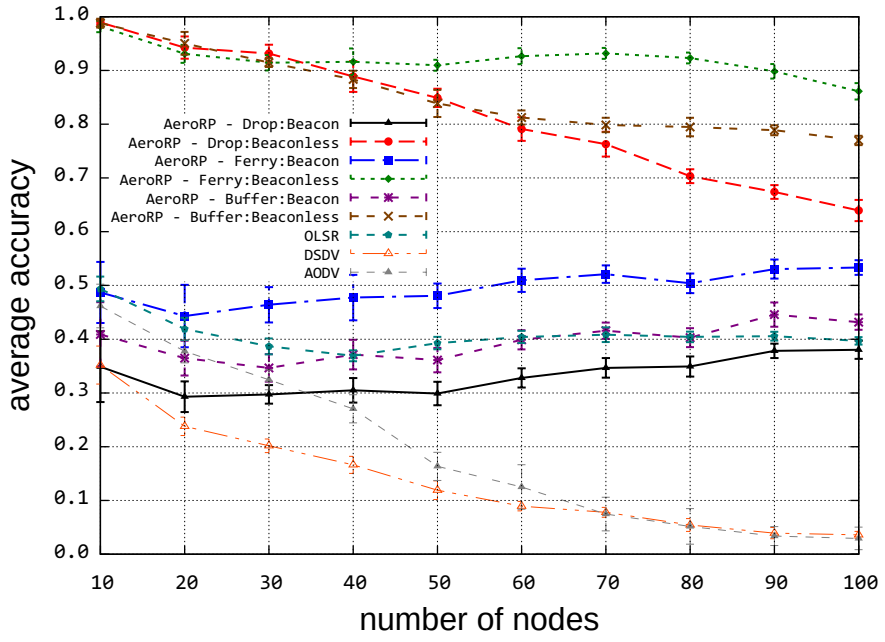

Figure A.12. Effect of node density on accuracy (200 ms/s - 1200 m/s)

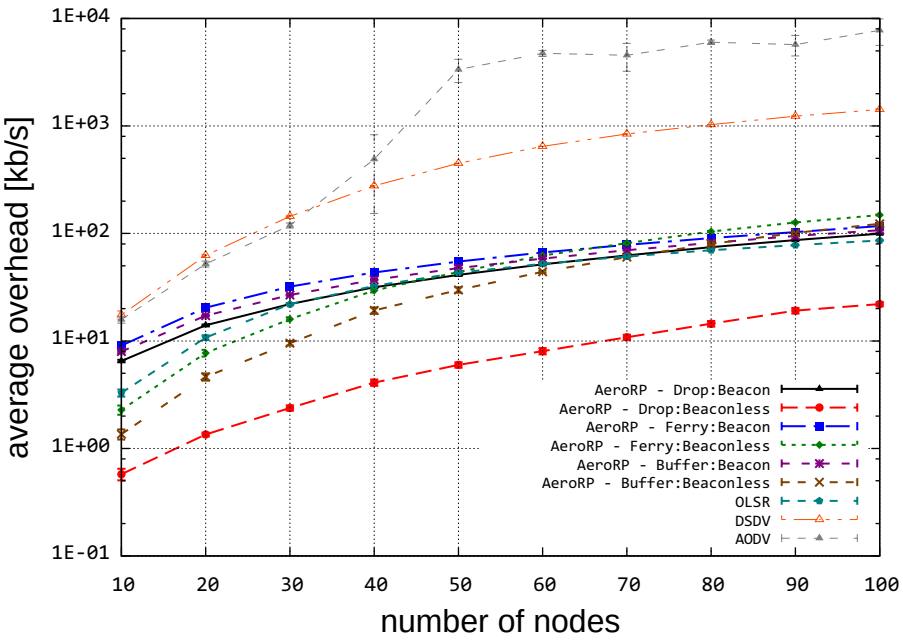

Figure A.13. Effect of node density on overhead (1200 m/s)

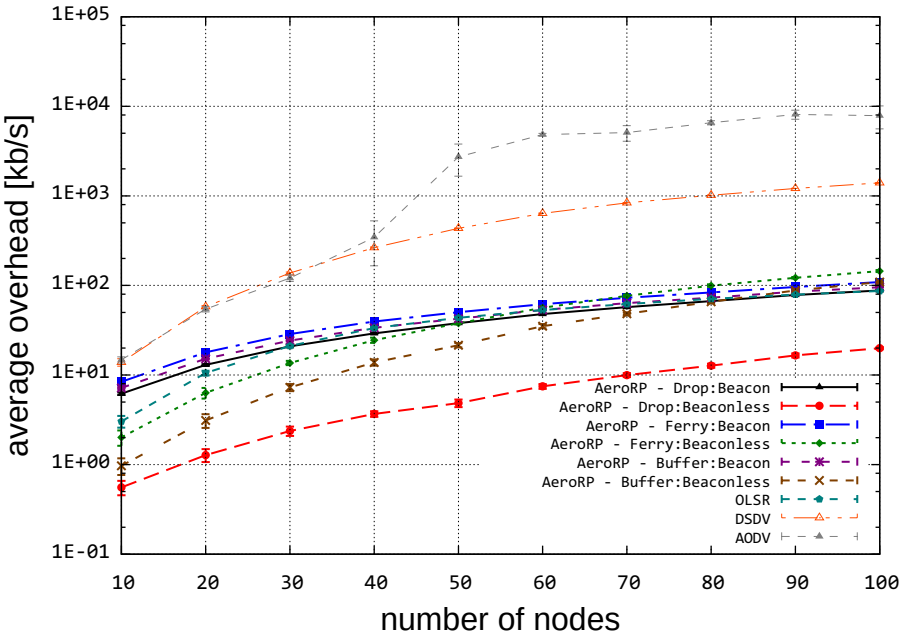

Figure A.14. Effect of node density on overhead (200 ms/s - 1200 m/s)

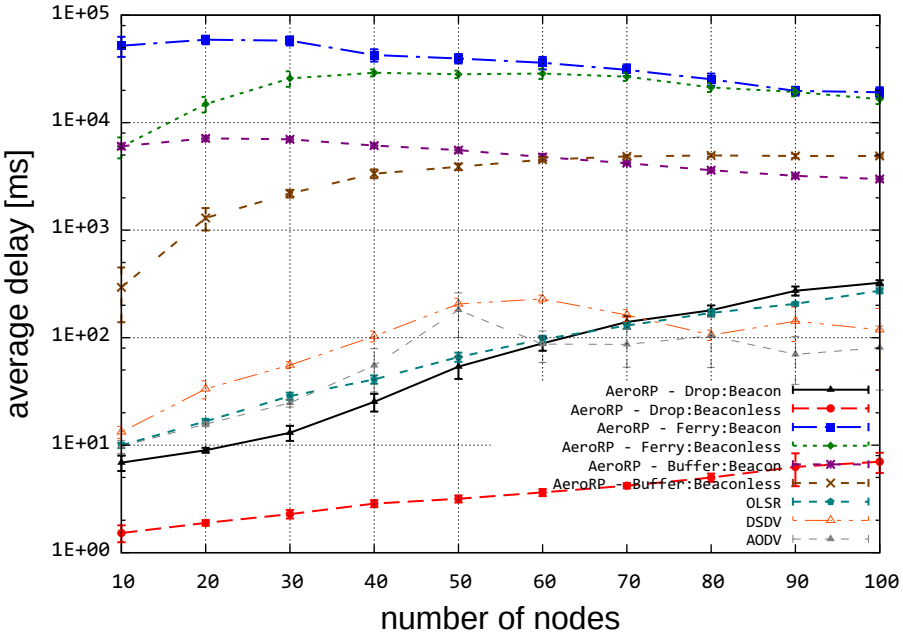

Figure A.15. Effect of node density on delay (1200 m/s)

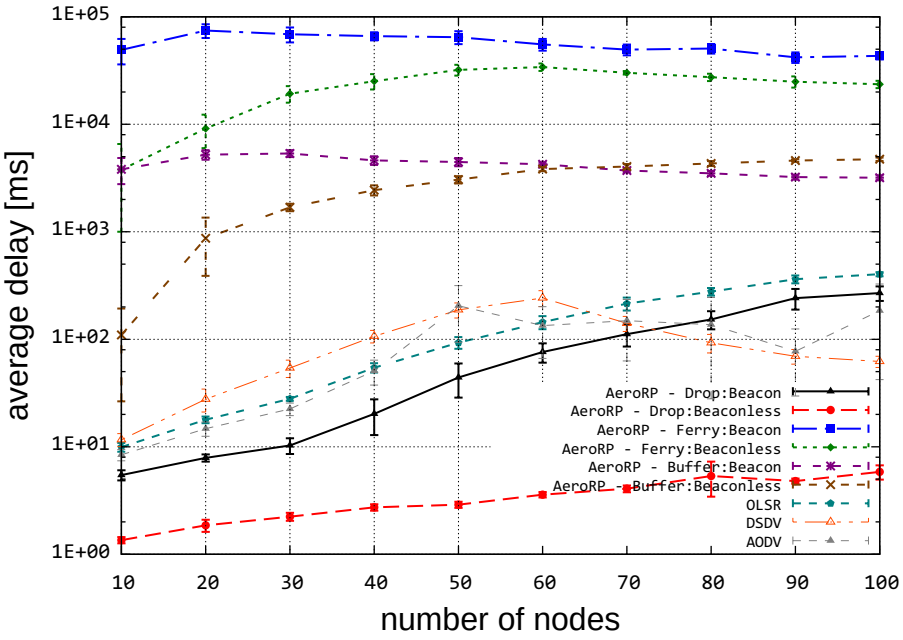

Figure A.16. Effect of node density on delay (200 ms/s - 1200 m/s)

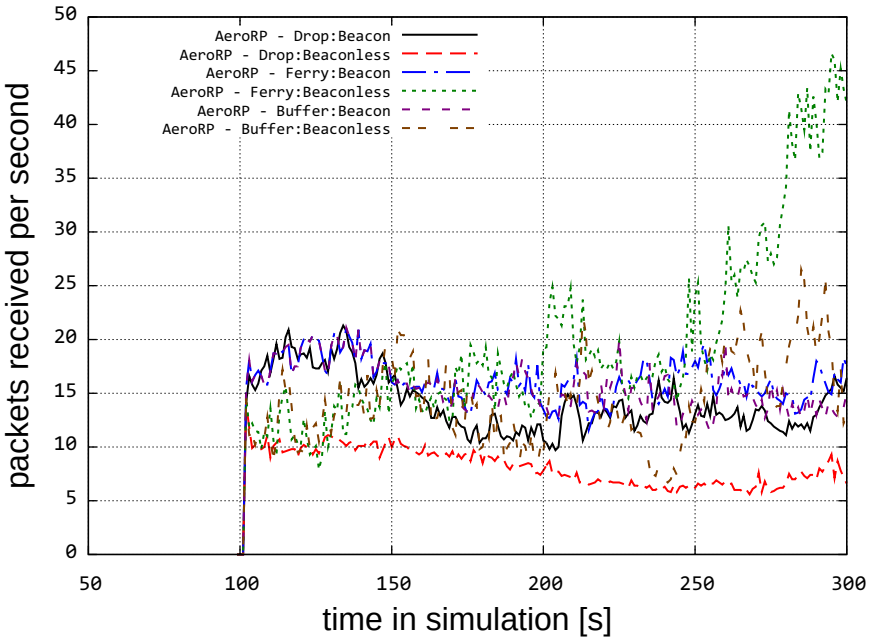

Figure A.17. Detailed look at startup (200 m/s - 60 nodes)

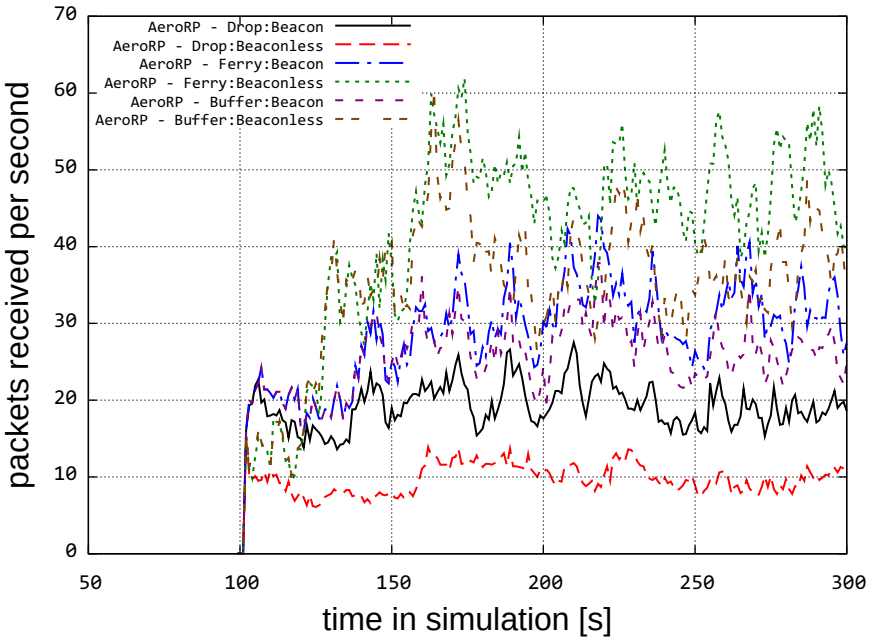

Figure A.18. Detailed look at startup (1200 m/s - 60 nodes)

References

[1] Justin P. Rohrer, Abdul Jabbar, Erik Perrins, and James P.G. Sterbenz. Cross-Layer Architectural Framework for Highly-Mobile Multihop Airborne Telemetry Networks. In *Proc. IEEE MILCOM2008*, San Diego, CA, USA, November 2008.

[2] Colin Lemmon, Siu Man Lui, and Ickjai Lee. Geographic Forwarding and Routing for Ad-hoc Wireless Network: A Survey. *International Conference on Networked Computing and Advanced Information Management*, 0:188–195, 2009.

[3] Stefano Basagni, Imrich Chlamtac, Violet R. Syrotiuk, and Barry A. Woodward. A Distance Routing Effect Algorithm for Mobility (DREAM). In *MobiCom '98: Proceedings of the 4th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, pages 76–84, New York, NY, USA, 1998. ACM.

[4] Brad Karp and H. T. Kung. GPSR: Greedy Perimeter Stateless Routing for Wireless Networks. In *MobiCom '00: Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, pages 243–254, New York, NY, USA, 2000. ACM.

[5] Alaa M. Al Tahan and Mohamed K. Watfa. A Position-based Routing Algorithm in 3D Sensor Networks. *Wireless Communications and Mobile Computing*, 2010.

[6] T. Camp, J. Boleng, and V. Davies. A Survey of Mobility Models for Ad Hoc Network Research. *Wireless Communication & Mobile Computing*, 2(5):483–502, 2002.

[7] iNET Working Group. http://www.inetprogram.org.

[8] iNET Needs Discernment Report, version 1.0. Central Test and Evaluation Investment Program (CTEIP), May 2004.

[9] iNET Technology Shortfalls Report, version 1.0. Central Test and Evaluation Investment Program (CTEIP), July 2004.

[10] iNET System Architecuture, version 2007.1. Central Test and Evaluation Investment Program (CTEIP), July 2007.

[11] S.M. Shatz J. Li. Toward Using Node Mobility to Enhance Greedy-forwarding in Geographic Routing for Mobile Ad Hoc Networks. In *The International Workshop on Mobile Device and Urban Sensing (MODUS 2008)*, April 2008.

[12] Roland Flury and Roger Wattenhofer. Randomized 3D Geographic Routing. In *27th Annual IEEE Conference on Computer Communications (INFOCOM), Phoenix, USA*, April 2008.

[13] The ns-3 Network Simulator. http://www.nsnam.org, July 2009.

[14] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot. Optimized Link State Routing Protocol for Ad Hoc Networks. In

Multi Topic Conference, 2001. IEEE INMIC 2001. Technology for the 21st Century. Proceedings. IEEE International, pages 62–68, August 2002.

[15] T. Clausen and P. Jacquet. Optimized Link State Routing Protocol (OLSR). RFC 3626 (Experimental), October 2003.

[16] Charles E. Perkins and Pravin Bhagwat. Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers. In *SIGCOMM '94: Proceedings of the Conference on Communications Architectures, Protocols and Applications*, pages 234–244, New York, NY, USA, 1994. ACM.

[17] C. Perkins, E. Belding-Royer, and S. Das. Ad Hoc On-Demand Distance Vector (AODV) Routing. RFC 3561 (Experimental), July 2003.

[18] D. Johnson, Y. Hu, and D. Maltz. The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4. RFC 4728 (Experimental), February 2007.

[19] David B. Johnson and David A. Maltz. Dynamic Source Routing in Ad Hoc Wireless Networks. In *Mobile Computing*, pages 153–181. Kluwer Academic Publishers, 1996.

[20] M. Mauve, A. Widmer, and H. Hartenstein. A Survey on Position-Based Routing in Mobile Ad Hoc Networks. *IEEE Network*, 15(6):30–39, 2001.

[21] M.G. de la Fuente and H. Ladiod. A Performance Comparison of Position-Based Routing Approaches for Mobile Ad Hoc Networks. *Vehicular Technology Conference*, pages 1–5, October 2007.

[22] Young-Bae Ko and Nitin H. Vaidya. Location-aided Routing (LAR) in Mobile Ad Hoc Networks. In *MobiCom '98: Proceedings of the 4th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, pages 66–75, New York, NY, USA, 1998. ACM.

[23] A. Capone, L. Pizziniaco, I. Filippini, and M.A.G. de la Fuente. A SiFT: an Efficient Method for Trajectory Based Forwarding. In *2nd International Symposium on Wireless Communication Systems*, pages 135 – 139, Sept. 2005.

[24] Abdul Jabbar and James P.G. Sterbenz. AeroRP: A Geolocation Assisted Aeronautical Routing Protocol for Highly Dynamic Telemetry Environments. In *Proceedings of the International Telemetering Conference*, Las Vegas, NV, October 26–29 2009.

[25] The Network Simulator: ns-2. `http://www.isi.edu/nsnam/ns/`, December 2007.

[26] James P. G. Sterbenz, Rajesh Krishnan, Regina Rosales Hain, Alden W. Jackson, David Levin, Ram Ramanathan, and John Zao. Survivable Mobile Wireless Networks: Issues, Challenges, and Research Directions. In *WiSE '02: Proceedings of the 1st ACM Workshop on Wireless Security*, pages 31–40, New York, NY, USA, 2002. ACM.

[27] Zhensheng Zhang. Routing in Intermittently Connected Mobile Ad Hoc Networks and Delay Tolerant Networks: Overview and Challenges. *IEEE Communications Surveys & Tutorials*, 8(1):24–37, March 2007.

[28] L. Pelusi, A. Passarella, and M. Conti. Opportunistic Networking: Data Forwarding in Disconnected Mobile Ad Hoc Networks. *IEEE Communications Magazine*, 44:134–141, 2006.

[29] Wenrui Zhao and Mostafa H. Ammar. Message Ferrying: Proactive Routing in Highly-Partitioned Wireless Ad Hoc Networks. In *FTDCS '03: Proceedings of the The Ninth IEEE Workshop on Future Trends of Distributed Computing Systems*, page 308, Washington, DC, USA, 2003. IEEE Computer Society.

[30] Ramesh Viswanathan, Jing (Tiffany) Li, and Mooi Choo Chuah. Message Ferrying for Constrained Scenarios. In *WOWMOM '05: Proceedings of the Sixth IEEE International Symposium on World of Wireless Mobile and Multimedia Networks*, pages 487–489, Washington, DC, USA, 2005. IEEE Computer Society.

[31] Utku Günay Acer, Shivkumar Kalyanaraman, and Alhussein A. Abouzeid. Weak State Routing for Large Scale Dynamic Networks. In *MobiCom '07: Proceedings of the 13th Annual ACM International Conference on Mobile Computing and Networking*, pages 290–301, New York, NY, USA, 2007. ACM.

[32] Mooi Choo Chuah and Wenbin Ma. Integrated Buffer and Route Management in a DTN with Message Ferry. *Journal of Information Science and Engineering*, 23(4):1123–1139, 2007.

[33] Mooi Choo Chuah, Peng Yang, Brian D. Davison, and Liang Cheng. Store-and-Forward Performance in a DTN. In *VTC Spring*, pages 187–191, 2006.

[34] E. Kuiper and S. Nadjm-Tehrani. Geographical Routing in Intermittently Connected Ad Hoc Networks. In *22nd International Conference on Advanced Information Networking and Applications*, pages 1690–1695, March 2008.

[35] Dan Broyles, Abdul Jabbar, and James P. G. Sterbenz. Design and analysis of a 3–D Gauss-Markov Mobility Model for Highly-Dynamic Airborne Networks. In *Proc. of the ITC*, San Diego, CA, October 2010. (to appear).

[36] Mun Choon Chan Shao Tao, A.L. Ananda. Spherical Coordinate Routing for 3D wireless Ad-hoc and Sensor Networks. In *33rd IEEE Conference on Local Computer Networks*, pages 144 –151, Oct. 2008.

[37] Robert L. Lidowski, Barry E. Mullins, and Rusty O. Baldwin. A Novel Communications Protocol using Geographic Routing for Swarming UAVs Performing a Search Mission. In *PERCOM '09: Proceedings of the 2009 IEEE International Conference on Pervasive Computing and Communications*, pages 1–7, Washington, DC, USA, 2009. IEEE Computer Society.

[38] Biao Zhou, Yeng-Zhong Lee, and M. Gerla. Direction Assisted Geographic Routing for Mobile Ad Hoc Networks. In *Military Communications Conference*, pages 1–7, Nov. 2008.

[39] Zhigang Jin, Ningxiao Yan, and Li Bing. Reliable On-Demand Geographic Routing Protocol Resolving Network Disconnection for VANET. In *5th International Conference on Wireless Communications, Networking and Mobile Computing*, pages 1–4, Sept. 2009.

[40] Sangsu Jung, Dujeong Lee, Sangyoon Yoon, Jaehwi Shin, Youngwoo Lee, and Jeonghoon Mo. A Geographic Routing Protocol Utilizing Link Lifetime and

Power Control for Mobile Ad Hoc Networks. In *FOWANC '08: Proceeding of the 1st ACM International Workshop on Foundations of Wireless Ad Hoc and Sensor Networking and Computing*, pages 25–32, New York, NY, USA, 2008. ACM.

[41] Quan Jun Chen, S.S. Kanhere, Mahbub Hassan, and Kun-Chan Lan. Adaptive Position Update in Geographic Routing. In *IEEE International Conference on Communications*, volume 9, pages 4046–4051, June 2006.

[42] Jongkeun Na and Chong kwon Kim. Glr: A novel geographic routing scheme for large wireless ad hoc networks. *Computer Networks*, 50(17):3434 – 3448, 2006.

[43] M. Iordanakis, D. Yannis, K. Karras, G. Bogdos, G. Dilintas, M. Amirfeiz, G. Colangelo, and S. Baiotti. Ad-hoc Routing Protocol for Aeronautical Mobile Ad-Hoc Networks. In *Fifth International Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP)*, 2006.

[44] N. Kato E. Sakhaee, A. Jamalipour. Aeronautical Ad Hoc Networks. In *IEEE Wireless Communications and Networking Conference*, volume 1, pages 246 –251, April 2006.

[45] Fabrice Tchakountio and Ram Ramanathan. Anticipatory Routing for Highly Mobile Endpoints. In *WMCSA '04: Proceedings of the Sixth IEEE Workshop on Mobile Computing Systems and Applications*, pages 94–101, Washington, DC, USA, 2004. IEEE Computer Society.

[46] Fabrice Tchakountio and Ram Ramanathan. Tracking Highly Mobile Endpoints. In *WOWMOM '01: Proceedings of the 4th ACM International Work-*

shop on Wireless Mobile Multimedia, pages 83–94, New York, NY, USA, 2001. ACM.

[47] Joseph Camp, Joshua Robinson, Christopher Steger, and Edward Knightly. Measurement Driven Deployment of a Two-tier Urban Mesh Access Network. In *MobiSys '06: Proceedings of the 4th International Conference on Mobile Systems, Applications and Services*, pages 96–109, New York, NY, USA, 2006. ACM.

[48] H. T. Friis. A Note on a Simple Transmission Formula. *Proceedings of the IRE*, 34(5):254–256, September 1946.

[49] IEEE 802.11-2007, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, June 2007.

[50] Wikipedia. atan2. `http://en.wikipedia.org/wiki/Atan2`, 2010. Online; accessed 04-April-2010.

[51] Egemen K. Çetinkaya and James P.G. Sterbenz. Aeronautical Gateways: Supporting TCP/IP-based Devices and Applications over Modern Telemetry Networks. In *International Telemetering Conference (ITC) 2009*, Las Vegas, NV, October 2009.

[52] James P.G. Sterbenz. AeroNP Specification, ANTP Project Document ANTP-NP-1.0, The University of Kansas, 2010.

[53] Yufei Cheng, Hemanth Narra, Abdul Jabbar, and James P.G. Sterbenz. Poster: Implementation of MANET Routing Protocols for ns-3. In *ITTC IAB Poster Session, The University of Kansas*, June 2010.

[54] RandomWaypoint Mobility Model. http://www.ittc.ku.edu/resilinets/media/60Nodes_27800TxRange_RandomWaypoint_mobility.mob4.avi, June 2010.

[55] GaussMarkov Mobility Model. http://www.ittc.ku.edu/resilinets/media/60Nodes_27800TxRange_GaussMarkov_mobility.mob4.avi, June 2010.

[56] Wikipedia. Student's t-distribution. http://en.wikipedia.org/wiki/Student's_t-distribution, 2010. Online; accessed 17-May-2010.

[57] Justin P. Rohrer, Erik Perrins, and James P.G. Sterbenz. End-to-End Disruption-Tolerant Transport Protocol Issues and Design for Airborne Telemetry Networks. In *International Telemetering Conference (ITC) 2008*, San Diego, CA, October 2008.